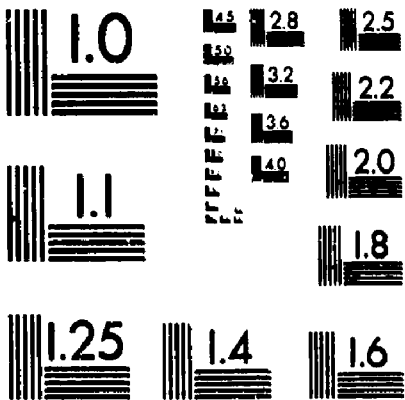


1 OF 2

30061

UNC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

NASA Contractor Report 166061

OPTIM

**Computer Program to Generate
a Vertical Profile Which
Minimizes Aircraft Fuel Burn
or Direct Operating Cost**

User's Guide

**Analytical Mechanics Associates, Inc.
Mountain View, California 94043**

**Prepared for
Langley Research Center
under Contract NAS1-15497**



**National Aeronautics and
Space Administration**

**Langley Research Center
Hampton, Virginia 23685**

May 1983

FOREWORD

The development of this computer program -- referred to as OPTIM -- was supported under NASA Contract No. NAS1-15497, by Langley Research Center, Hampton, Virginia. The project technical monitors were Samuel A. Morello, Kathy H. Samms, and Robert E. Shanks. At AMA, Inc., the project manager was John A. Sorensen, with engineering support provided by Mark H. Waters. The project programmers were Marianne N. Kidder, Quyen T.L. Nguyen, and Leda C. Patmore.

OPTIM is an extensive modification of an original program developed by Heinz Erzberger and Homer Q. Lee of NASA Ames Research Center. Technical discussions with Dr. Erzberger and Mr. Lee are gratefully acknowledged. Also, suggestions for program improvement by Ms. Samms and other members of the NASA Langley Research Center staff have been greatly appreciated.

This User's Guide describes the program input, program output, and general organization. Appendix A presents the technical material upon which the program is based. Appendix B presents a brief explanation of each of the program subroutines.

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. INPUT DESCRIPTION	3
III. OUTPUT DESCRIPTION	13
IV. PROGRAM ORGANIZATION	27
APPENDIX A. TRAJECTORY OPTIMIZATION USING THE ENERGY STATE METHOD . .	47
Theoretical Principles	47
Application to Aircraft Profile Optimization Using the Energy State Approximation	50
Some Mechanization Details of the Computer Program	60
APPENDIX B. OPTIM SUBROUTINE DESCRIPTION.	71
REFERENCES	115

PRECEDING PAGE BLANK PAGE

I

INTRODUCTION

This document is a technical description and a user's guide for a computer program -- called OPTIM -- which is used to design near optimum vertical profiles for turbojet powered aircraft. Specifically, the program generates a profile of altitude, airspeed, and flight path angle as a function of range between a given set of origin and destination points for particular models of transport aircraft provided by NASA. Inputs to the program include the vertical wind profile, the aircraft takeoff weight, the costs of time and fuel, certain constraint parameters and control flags. The profile can be near optimum in the sense of minimizing: (a) fuel, (b) time, or (c) a combination of fuel and time (direct operating cost (DOC)). The user can also, as an option, specify the length of time the flight is to span. The theory behind the technical details of this program appears in Appendix A.

OPTIM is an adaptation and an extensive modification of another program developed by Dr. Heinz Erzberger and Mr. Homer Lee of NASA Ames Research Center for the IBM 360 computer. The present program has been modified to operate on the NASA Langley Research Center CDC 6600 and PDP 11/70 computers.

A companion program, has been constructed which takes the output of OPTIM as input. This companion program -- called TRAGEN -- simulates an aircraft steered to follow the profile generated by OPTIM. The user's guide for TRAGEN appears as a separate document [1].

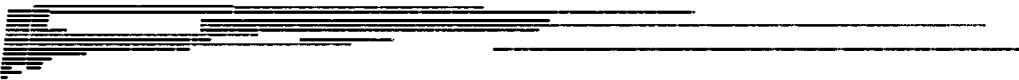
OPTIM has the following applications:

1. It can determine how much fuel consumption and operating costs can be reduced by flying an optimum path rather than a reference trajectory specified in the pilot's handbook.
2. It serves as a benchmark for evaluating sub-optimum algorithms.
3. It can be incorporated into an airline's flight planning system.

ORIGINAL FILED
OF PUCK COUNCIL

4. It can be incorporated into advanced automatic air traffic control software.
5. It serves as the basis for the design of an advanced flight management system.
6. It can be used to assess the advantage of alternate engines or aerodynamic changes on air transport operating costs.

The chapters in this document are organized as follows:

1. Chapter II explains the meaning of input variables required to run the program.
 2. Chapter III explains the meaning of program output variables and options.
 3. Chapter IV presents a subroutine layout and flowcharts explaining the basic organization of the program.
 4. Appendix A briefly summarizes the theory behind the OPTIM program.
 5. Appendix B explains the purpose of each of OPTIM's subroutines and functions.
-
-
- 

II

INPUT DESCRIPTION

To run OPTIM requires up to six input cards and up to three designated data files. The meanings of the variables on the input cards are given first. The program uses Unit 5 as the card input source.

Card 1

This card is the header that appears at the beginning of the run. The input has an 8A10 format.

Card 2

This card consists of ten integer variables used as flags to control the operation of the program. The input numbers are right-justified and have a 20I4 format. They are:

ICTAB ICOUT IPRINT IVPI IWIND ICALT ISPLMT IGRAF IAC
IDCC.

The meaning of each of these variables is as follows:

ICTAB In generating the climb and descent portions of the profile, the program uses a table of optimum cruise conditions. This table, called the "Cruise Table", gives the optimum cruise altitude and airspeed as a function of cruise weight. The program uses the results of this table to produce boundary conditions for the optimization process. The program has the option of generating a new cruise table or using a cruise table generated by a previous run. A new table should be generated either if a different aircraft model (aerodynamic and engine data), wind profile, aircraft heading, temperature variation, or cost penalty on time or fuel is used. Values of ICTAB are:

ORIGINAL PAGE IS
OF POOR QUALITY

ICTAB = 0: Causes the calculation of a new cruise table.

ICTAB = 1: Uses an existing cruise table to be read in from Unit 8.

ICTAB = 2: Time-of-arrival fixed (create new cruise tables).

Creating a cruise table when cruise altitude is free to vary is time consuming. (See input option ICALT). Free altitude optimization runs may take up to ten times as long as fixed altitude runs because of this calculation. It is therefore advisable to save (option ICOUT) and reuse old cruise tables whenever possible for free altitude runs (ICALT = 0 or 3). It is not possible, however, to save or reuse fixed altitude or time-of-arrival cruise tables.

ICOUT This option controls the writing of a cruise table on Unit 8. It is only operative during free cruise altitude runs. The user is responsible for saving Unit 8 output in a permanent file after completion of a run, for rewinding it between runs, and for recovering before a new run. Values of ICOUT are;

ICOUT = 0; Do not write cruise table on Unit 8.

ICOUT = 1; Do write cruise table on Unit 8.

IPRINT This flag is used to control the amount of printout during the program computation process. Values of IPRINT are:

IPRINT = 0: Normal mode (see Chapter III, for output description)

IPRINT = 1: Extra printout included. This produces detailed output useful for debugging cruise table calculations. Some familiarity with the program is necessary in order to use this output.

IPRINT = 2: Less printout (input, cruise table summary, summary of flight parameters (time, distance, fuel) for each climb and descent iteration, optimum profile summary).

IPRINT = 3: Minimum printout (input, cruise table summary, summary of flight parameters for optimum profile.)

ORIGINAL PAGE IS
OF POOR QUALITY

IVPI To determine the optimum profiles, the user has the option of using airspeed as the control variable (with thrust fixed) or both airspeed and thrust as control variables. If thrust is fixed, it is set to the maximum value for climb and to the minimum value (idle throttle) during descent. Using two controls gives slightly better performance trajectories in terms of lower overall cost. However, for short-haul operations, the profile shape can be substantially different, as shown in Appendix A. Values of IVPI are:

IVPI = 0: Optimize using only airspeed as a control.

IVPI = 1: Optimize using both airspeed and thrust as controls.

IWIND An arbitrary wind profile can be read in on Unit 7. It gives the wind speed and heading as a function of altitude (above sea level). Values of IWIND are:

IWIND = 0: No wind used.

IWIND = 1: Input wind profile varies with altitude but is constant over the entire range of flight. (See description of input wind data, p. 9).

IWIND = 2: Separate wind profiles are input for climb, cruise and descent portions of the flight.

ICALT The program has the option to generate a three-part profile (consisting of climb-cruise-descent) or a two-part profile (consisting of cruise-descent) with or without fixed cruise altitude. An option to add a step climb segment during cruise is also available. The values of ICALT are:

ICALT = 0: Three-part profile with a free cruise altitude.

ICALT = 1: Three-part profile with fixed cruise altitude. With this option, if the input range is not long enough to allow the aircraft to climb to and descend from the input cruise altitude, a feasible altitude will be sought. The final altitude will be some multiple of 2000 feet less than the input altitude.

ICALT = 2: Three-part profile with fixed cruise altitude and step climb. The program will assume a 4000 foot climb at maximum thrust after attainment of the fixed cruise altitude. The optimum

ORIGINAL PAGE IS
OF POOR QUALITY

distance into cruise at which the step climb starts is solved for along with the other optimization variables.

ICALT = 3: Two-part profile with free cruise altitude. Initial cruise weight and range-to-go are input. OPTIM solves for the optimum initial cruise altitude and airspeed in addition to the rest of the profile.

ICALT = 4: Two-part profile with fixed cruise altitude. Initial cruise weight, range-to-go, and altitude are input. OPTIM solves for the optimum cruise airspeed with altitude fixed.

ICALT = 5: Two-part profile with fixed cruise altitude and step climb. This option is similar to ICALT = 2, except that the flight starts at an input initial cruise weight and altitude.

ISPLMT This flag allows the user to remove the 250 kt indicated airspeed limit below 10,000 ft. Values of ISPLMT are:

ISPLMT = 0: No V_{IAS} limit for altitude below 10,000 ft.

ISPLMT = 1: 250 kt V_{IAS} limit below 10,000 ft (nominal).

ISPLMT = 2: 250 kt V_{IAS} limit below 10,000 ft for descent only.

IGRAF This flag controls the output of a data set containing the optimum profile which is used for generating graphs and as input to the TRAGEN program. The data set is output on UNIT 11. Values of IGRAF are:

IGRAF = 0: Do not write an output data set.

IGRAF = 1: Write a data set using Unit 11.

IGRAF = 2: Write a data set using Unit 11. Printer plots are drawn for the four variables: Mach no., flight path angle, altitude and fuel versus range.

IGRAF > 2: A data set is written on unit 11. Printer plots are generated for all variables.

IAC This flag is used to select which aircraft model to use to generate the optimum flight profile. Current values of IAC are:

OPTIONAL TABLES
OF POOR QUALITY

IAC = 2: Medium-range three-engine jet transport aircraft.

IAC = 3: Medium-range two-engine jet transport aircraft.

IDCC Descent may be constrained to a constant sink rate to maintain cabin pressurization at a given differential until cabin pressure reaches sea level pressure. At present this differential is set at 10 psi (pounds per square inch). The constant sink rate is 500 ft/min. Values of IDCC are:

IDCC = 0: Unconstrained descent

IDCC = 1: Constrained descent.

Card 3 (Optional)

This card has four real variables with format 8F10.2. It is used only when a new cruise table is to be generated (ICTAB = 0 or 2 on Card 1). The variables are:

FC TC DTEMPK PSIG.

The meanings of these variables are as follows:

FC This is the cost of jet fuel in \$/lb (e.g., 0.15).

TC This is the cost of time in \$/hr (e.g., 600.00). Both FC and TC are used in the cost function which is minimized by the program. If time-of-arrival is fixed, this variable is ignored.

DTEMPK This is the temperature variation from standard atmospheric conditions in degrees Kelvin.

PSIG This is the aircraft ground heading in degrees. It is used along with the wind heading to compute aircraft heading with respect to the airmass.

Card 4 (Optional)

This card has three real variables with format 8F10.2. It is used only when a new cruise table is to be generated (ICTAB = 0 or 2 on Card 1). The variables are:

WEIGHT WN DEW.

ORIGINAL PAGE IS
OF POOR QUALITY

The meanings of these variables are as follows:

WEIGHT This is the maximum value that the weight of the aircraft can be in pounds (e.g., 150,000 lb). The first cruise table will be generated at this weight.

WN This is the minimum value that the weight of the aircraft can be in pounds (e.g., 110,000 lb).

DEW This is the incremental cruise weight in pounds between each table (e.g., 5,000 lb). Starting with WEIGHT, a cruise table will be generated for cruise weights of WEIGHT, WEIGHT-DEW, WEIGHT-2 DEW, etc., down to WN. A maximum of ten cruise tables can be generated because of internal program array dimensioning.

Card 5

This card has five real variables with format 8F10.2. The variables are:

WTO RANGE DEIN HCRUZ TEND.

The meanings of these variables are as follows:

WTO Aircraft takeoff weight for three-part profile or initial weight for two-part profile in pounds (e.g., 136,000 lb).
NOTE: This value must be less than or equal to WEIGHT of Card 4.

RANGE Range-to-go in nautical miles (e.g., 200 n.mi.).

DEIN Incremental specific energy in feet between points on the optimum profile. As explained later in Appendix A, the program uses an energy state method to generate the optimum trajectory. Specific energy is the independent variable, and varying the size of DEIN affects the smoothness and accuracy of the generated profile. If DEIN is input as 0., it is set equal to a nominal value of 500 feet.

HCRUZ This is the value of the fixed-cruise altitude (in feet)
that is used when ICALT is set other than 0 or 3.

TEND This is the value of the desired time-of-arrival (in seconds)
that is used when ICTAB is set to 2.

Card 6

This card has four real variables with format 8F10.2. The variables are:

HTO Initial aircraft altitude in feet.

VTO Initial indicated airspeed in knots.

HOLNDG Final aircraft altitude in feet.

VOLNDG Final indicated airspeed in knots.

OF POOR QUALITY

Note that if a cruise table already exists, only Cards 1, 2, 5, and 6 are required. An example of cards 2 through 5 is:

0	0	3	0	0	2	1	0	3		
	0.15		600.			0.		90.		
100000.			70000.		5000.					
100000.			1000.		500.		33000.		5000.	
	0.		210.		0.		210.			

In addition to the card input, there are up to three data sets that may be used by the program. These are:

Unit 7 - Wind Data (Optional)

This data set is used when IWIND is set to 1 or 2. The input consists of the magnitude of the wind and the direction of its source as a function of altitude. The data format is (3F5.0, I2).

If IWIND = 1, a single wind profile applicable to the entire flight is read in. This profile consists of a set of n cards. Each card has four variables.

HWIND(I) PSIW(I) VW(I) IE

There is one card for each $I=1,2,\dots,N$, where N is the number of altitudes used for a given wind profile. The meanings of these variables are:

ORIGINAL PAGE IS
OF POOR QUALITY

- HWIND(I) Beginning (lowest) altitude at which direction PSIW(I) and magnitude VW(I) apply. The program will interpolate for values of PSIW and VW when using altitudes between HWIND(I) and HWIND(I+1).
- PSIW(I) Direction of the wind vector source in degrees (i.e., 270° represents a wind from the West).
- VW(I) Magnitude of the wind vector in knots.
- IE End-of-wind-table indicator. If IE = 0 the program will expect to read further wind data. If IE = 1 the program assumes that a complete wind table has been read in. Note that when IE = 1, the corresponding altitude should be equal to or greater than any altitude the aircraft is expected to reach.

If IWIND = 2, three wind profiles are read in, one each for climb, cruise, and descent (in that order). Each profile is as described under IWIND = 1. Each profile must end with a non-zero value for IE.

Unit 8 - Cruise Table Data

Data are generated by the program and written on Unit 8 when ICOUT is set to 1. Most of this data are also output to Unit 6 when IPRINT = 0. Data generated during a previous run are read in from Unit 8 when ICTAB is set to 1; in this case, only the summary of the cruise table is output to Unit 6. The cruise table data that are written on Unit 8 are as follows: (the Format is 8E15.7 unless specified otherwise):

- Line 1: FC, TC, DTEMPK, PSIA obtained from Card 2.
- Line 2: WUSE - aircraft cruise weight obtained in going from WEIGHT to WN in steps of DEW as defined by Card 3.
- Line 3: HALT altitude - 10,000 to 40,000 ft (or to H_{max}) in steps of 1,000 ft.
- | | | | |
|---------|------------------------------|---|---|
| EMAKIAS | minimum drag airspeed - kt | } | Optimum cruise conditions at altitude H for minimum DOC per n.mi. |
| FBIAS | maximum cruise airspeed - kt | | |
| OPMIAS | indicated airspeed - kt | | |

OPMTAS	true airspeed - kt	} Optimum cruise conditions at altitude H for minimum DOC per n.mi.
OPMACH	Mach	
EPRS	EPR setting	
FDTOPT	cost in \$/n.mi. (λ)	
FUELDT	fuel flow rate for FDTOPT	

These lines are repeated for altitude varying from H_{\min} to H_{\max} .

Line 4: ENDATA -10^6 , code for end of a weight set.

Line 5: HOPT optimum altitude where λ (FDTOPT from line 3) is minimum

OPTMAK optimum Mach (at HOPT)

OPTIAS optimum indicated airspeed - kt

OPTTAS optimum true airspeed - kt

EMCOST minimum - \$/n.mi.

EPRT optimum EPR setting (at HOPT)

EOPT optimum cruise energy (at HOPT)

FUELDT optimum fuel flow rate.

An example of lines 1-5 appears as an output table in Chapter III.

Lines 2-5 are repeated for each weight, from WEIGHT to WN in steps of DEW.

Line 6: ENDATA -10^6 , code for end of weight tables.

Line 7: IWMAX (Format 14) number of cruise tables
(=(WEIGHT-WN)/DEW) + 1.)

Line 8: (DLLDEE(1,J), DLLDEE(2,J), J=1, IWMAX) - the coefficients of the derivative of the optimum cost EMCOST as a function of energy for each weight table.

Line 9:	WS(1)	cruise weight - lb	} Repeated for I = 1, IWMAX
	EOPTS(1)	optimum cruise energy for WS(1) - ft	
	EMSTAR(1)	optimum Mach	
	HSTARS(1)	optimum altitude - ft	
	PISTRS(1)	optimum EPR	
	ELAMBS(1)	minimum cost - \$/n.mi.	
	VTASOP(1)	optimum true airspeed - kt	
	FUELFI(1)	fuel flow rate at PISTRS(1) - lb/hr.	

OUTPUT DESCRIPTION

ORIGINAL PAGE IS
OF POOR QUALITY

The output of OPTIM is lengthy, in tabular form, and generally self-explanatory. This section presents one sample of each of several different types of output written to Unit 6. The quantity of this output is controlled by the input flag IPRINT. Unit 8 (input flag ICOUT) and Unit 11 (flag IGRAF) can be used to store data sets for later use if desired.

A summary of the flags and input variables is printed at the beginning of each run. Table 1 is an example of this output.

The next table output is the vertical wind profile. This is printed when there is a non-zero wind, and the flag IWIND is set to 1 or 2. An example is presented here as Table 2.

The next output tables are the cruise tables. These are printed if they are generated as part of the run. That is, if the flag ICTAB is set to 0 or 2, new cruise tables are computed based on other input data. Table 3a is an example of a cruise table for a cruise weight of 100000 lb. Each column of this table is as defined on pp. 10-11 (in the explanation of Line 3, Unit 8). The format of the printed version varies slightly from the format of the data set output to Unit 8.

If the cruise tables are not generated as part of the run, but read in from Unit 8, then a table such as Table 3b is printed. This table presents the derivative of the cost per n.mi. (referred to as λ) as a function of change in the cruise energy. The coefficients of this derivative are shown for cruise weights from WN to WEIGHT in steps of DEW. (See input description, page 8).

Following Tables 3a or 3b is a summary of the optimum cruise conditions for each of the distinct cruise weights in terms of minimum direct operating costs. This table is obtained by interpolating the conditions of each cruise table, such as Table 3a, to determine the optimum altitude/airspeed/power setting combination for a given cruise weight. An example of this summary table is shown as Table 4.

ORIGINAL PAGE IS OF POOR QUALITY

Table 2. Vertical Wind Magnitude and Direction as a Function of Altitude. PSIW is direction of wind source.

OWIND DATA CLIMB			
ALT(FT),	VW(KNOTS),	VW(FT/SEC),	PSIW(DEG)
0.	0.00	0.00	90.
40000.	25.00	42.20	90.
OWIND DATA CRUISE			
ALT(FT),	VW(KNOTS),	VW(FT/SEC),	PSIW(DEG)
0.	0.00	0.00	90.
40000.	50.00	84.39	90.
OWIND DATA DESCEND			
ALT(FT),	VW(KNOTS),	VW(FT/SEC),	PSIW(DEG)
0.	0.00	0.00	90.
40000.	100.00	168.78	90.
O AIRCRAFT HEADING =		90.	DEG

After the summary table is printed, an estimate is made of the initial cruise weight (takeoff weight minus fuel burned during climb). This value is then used to generate a new line of the cruise table based on interpolation. The result is printed out and is shown here as Table 5. The variable LAMBDA from Table 5 is used as a key search variable in constructing the optimum trajectory. This is explained in more detail in Appendix A.

After the first five tables are printed, the program goes into an interactive search process to compute the optimum climb and descent portions of the trajectory. The number of iterations varies and depends on the range to be flown, whether a two or three-part trajectory is solved for, whether optimization uses airspeed or airspeed/thrust as controls, and how close to the desired range the final trajectory is supposed to be for convergence. An example of the output given in tabular form for the climb trajectory is shown in Table 6a.

The descent trajectory is computed backwards in time starting with the estimated landing weight and ending with the final cruise conditions. An example of this descent trajectory is shown as Table 6b.

A summary of the climb, cruise, and descent segments of a three-part trajectory in terms of fuel, distance, time, cost and cost/n.mi. is printed finally in the form of Table 7a. This table is the essence of the program's output. It shows how close to the desired range the trajectory came, and what total cost resulted. This table is used to rapidly compare

ORIGINAL PAGE IS
OF POOR QUALITY

Table 3a. One Page of the Cruise Table for Cruise Weight of 100,000 lb.
(IPRINT = 0 or 1 only)

1 0	AIRCRAFT CRUISE WT = 100000. LBS		MINIMUM COST/DISTANCE		MINIMUM FUEL		MINIMUM FUEL RATE	
	ALT FT	HM DRAG KIAS	SPEED KIAS	MACH	EPR	PWR SLIG	FUEL LB/HR	FUEL LB/HR
	10000.	215.	425.	.641	1.452	4.32	7412.	7412.
	11000.	216.	426.	.642	1.472	4.25	7385.	7385.
	12000.	216.	417.	.642	1.484	4.17	7242.	7242.
	13000.	216.	411.	.640	1.498	4.10	7109.	7109.
	14000.	216.	404.	.644	1.498	4.04	6762.	6762.
	15000.	217.	400.	.660	1.500	3.98	6452.	6452.
	16000.	217.	394.	.653	1.500	3.92	6123.	6123.
	17000.	217.	389.	.656	1.511	3.88	5985.	5985.
	18000.	216.	384.	.677	1.529	3.83	5967.	5967.
	19000.	218.	379.	.677	1.548	3.79	5859.	5859.
	20000.	217.	372.	.683	1.564	3.75	5832.	5832.
	21000.	217.	366.	.694	1.584	3.71	5783.	5783.
	22000.	218.	360.	.702	1.603	3.67	5694.	5694.
	23000.	218.	353.	.708	1.621	3.63	5544.	5544.
	24000.	219.	346.	.711	1.637	3.59	5462.	5462.
	25000.	219.	339.	.715	1.655	3.56	5396.	5396.
	26000.	219.	332.	.720	1.677	3.53	5316.	5316.
	27000.	220.	324.	.724	1.698	3.51	5213.	5213.
	28000.	220.	318.	.725	1.719	3.49	5136.	5136.
	29000.	220.	310.	.727	1.744	3.48	5037.	5037.
	30000.	221.	302.	.727	1.768	3.47	4970.	4970.
	31000.	220.	295.	.729	1.797	3.46	4914.	4914.
	32000.	217.	288.	.730	1.830	3.46	4843.	4843.
	33000.	214.	281.	.729	1.864	3.47	4770.	4770.
	34000.	212.	273.	.726	1.901	3.48	4776.	4776.
	35000.	212.	264.	.726	1.945	3.52	4812.	4812.
	36000.	213.	257.	.725	1.998	3.56	4899.	4899.
	37000.	213.	247.	.724	2.059	3.61	5000.	5000.
	38000.	214.	237.	.724	2.124	3.68	5179.	5179.
	39000.	215.	224.	.723	2.199	3.77	5153.	5153.
	39497.	215.	216.	.710	2.229	3.84	5198.	5198.
	39624.	214.	214.	.713	2.241	3.84	5222.	5222.
	39687.	216.	216.	.714	2.247	3.85	5234.	5234.
	39718.	216.	216.	.714	2.250	3.85	5241.	5241.
	39733.	216.	216.	.714	2.252	3.85		

MIN (FDOT/U) = 3.441 LB/HR
MIN FUEL RATE = 4941. LB/HR

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

OPTIMIZING FUEL/DISTANCE:

OPT ALT = 31510. FT
CRUISE POWER SETTING = 1.8139 EPR

EPR

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

KTAS

CRUISE TABLE SUMMARY

Table 3b. Derivative of the Cruise Cost with Respect to Cruise Energy. (IPRINT = 0 or 1 only)

$$D(\text{LAMBDA})/DE = A E + B \text{ IN S/NM}^{+2}$$

CRUISE WT	A	B
150000.	.1339257E-04	-.6511878E+00
145000.	.1299522E-04	-.6502478E+00
140000.	.1342047E-04	-.6792299E+00
135000.	.1298901E-04	-.6735543E+00
130000.	.1228388E-04	-.6573249E+00
125000.	.1269218E-04	-.6756102E+00
120000.	.1036897E-04	-.5991612E+00
115000.	.8721075E-05	-.5484596E+00
110000.	.7292038E-05	-.5016609E+00

Table 4. Cruise Table Summary.

CRUISE TABLE SUMMARY								
OCRUISE WT	OPT E	OPT MACH	OPT H	PWR SET	LAMBDA	KTAS	FUEL FLOW	
LB	FT		FT	EPR	S/NM		LB/HR	
0 100000.	40874.	.7252	33000.	1.8588	3.130	421.90	4794.73	
0 95000.	40880.	.7255	33000.	1.8204	3.061	422.06	4605.92	
0 90000.	40907.	.7268	33000.	1.7865	2.997	422.80	4442.08	
0 85000.	40883.	.7256	33000.	1.7522	2.939	422.14	4264.84	
0 80000.	40918.	.7273	33000.	1.7243	2.885	423.08	4132.03	
0 75000.	40950.	.7287	33000.	1.6987	2.836	423.93	4010.24	
0 70000.	40954.	.7289	33000.	1.6737	2.794	424.05	3893.52	

Table 5. Interpolation of Cruise Table for Estimated Weight at Top of Climb. IPRINT = 3.

WEIGHT AT BEGINNING OF CRUISE							
OCRUISE WT	OPT E	OPT MACH	OPT H	LAMBDA	KTAS	FUEL FLOW	
LB	FT		FT	S/NM		LB/HR	
0 97460.	40877.	.7254	33000.	3.094	421.98	4699.81	

ORIGINAL PAGE IS
OF POOR QUALITY

Table 6. Optimum Climb Trajectory. (IPRINT = 0 or 1 only).

AIRCRAFT TAKE OFF WT = 136000. LBS, CRUISE ENERGY = 11431. FT, INITIAL CRUISE WT = 134934. LBS									
INITIAL ALT (FT), SPEED (KIAS) C. 216.									
FUEL COST(1/LB) .0625 TIME COST(1/HR) = 500.00 TEMP VAR (DEG K) = 0.00 LAMDA = 3.3315/M									
CLIMB OPTIMIZATION:									
ENERGY ALTITUDE FT	MACH	VIAS KNOT	VIAS KNOT	EDDY FT/SEC	FPTH DEG	TIME H:M:S	DIST N MILE	FUEL USED PWR SETG Lb	COST/E \$/E FT
1904.	0.	.314	207.	56.66	0.00	0: 0: 0	0.000	53.	1.803
2032.	2.	.324	217.	58.57	.02	0: 0: 17	.560	104.	1.801
2532.	6.	.361	243.	64.00	.68	0: 0: 29	.991	152.	1.792
3032.	260.	.378	254.	65.28	4.73	0: 0: 32	1.509	198.	1.780
3532.	744.	.381	253.	64.57	8.40	0: 0: 40	2.041	245.	1.766
4032.	1204.	.384	253.	63.90	8.00	0: 0: 48	2.580	291.	1.801
4532.	1660.	.387	253.	63.38	7.89	0: 0: 56	3.128	337.	1.806
5032.	2126.	.390	253.	62.79	7.76	0: 1: 4	3.684	384.	1.811
5532.	2587.	.394	253.	62.21	7.64	0: 1: 12	4.248	430.	1.816
6032.	3046.	.397	253.	61.62	7.52	0: 1: 20	4.820	476.	1.821
6532.	3505.	.400	253.	61.04	7.40	0: 1: 28	5.401	522.	1.826
7032.	3963.	.404	253.	60.51	7.29	0: 1: 36	5.991	569.	1.831
7532.	4420.	.407	254.	59.95	7.17	0: 1: 44	6.588	616.	1.837
8032.	4877.	.411	254.	59.40	7.06	0: 1: 53	7.195	667.	1.842
8532.	5333.	.414	254.	58.84	6.94	0: 2: 2	7.817	706.	1.849
9032.	5788.	.417	255.	58.71	6.83	0: 2: 10	8.436	729.	1.851
9532.	6223.	.421	255.	58.42	6.73	0: 2: 18	9.073	752.	1.854
10032.	6676.	.423	255.	58.13	6.71	0: 2: 26	9.754	774.	1.856
10532.	7129.	.426	255.	57.56	6.66	0: 2: 34	10.473	797.	1.859
11032.	7582.	.428	255.	57.27	6.62	0: 2: 42	11.239	820.	1.861
11532.	8032.	.430	255.	56.98	6.57	0: 2: 50	12.044	843.	1.864
12032.	8481.	.432	255.	56.69	6.52	0: 3: 0	12.899	866.	1.866
12532.	8932.	.434	255.	56.39	6.48	0: 3: 8	13.702	889.	1.869
13032.	9382.	.436	255.	56.10	6.43	0: 3: 16	14.562	911.	1.871
13532.	9832.	.437	255.	55.80	6.37	0: 3: 24	15.439	934.	1.874
14032.	10282.	.437	255.	55.50	6.30	0: 3: 32	16.370	957.	1.876

Table 6b. Optimum Descent Trajectory (in Backwards Time).
(IPRINT = 0 or 1 only).

AIRCRAFT LANDING WT - 135444. LWS, CRUISE ENERGY - 11431. FT															
FINAL ALTITUDE, SPEED (KIAS) - 0. 210.															
DESCEND OPTIMIZATION:															
ENERGY	ALTITUDE	MACH	VIAS	VIAS	VTAS	EDOT	FPTH	TIME	DIST	FUEL USED	PWR SETS	COST/E	SUN COST/E	S/E FT	S/E FT
FT	FT	NO	KNOT	KNOT	FT/SEC	FT/SEC	DEG	HR:MM:SS	N MTLF	Lb	EPH	\$/E FT	\$/E FT		
1904.	0.	.314	210.	207.	-21.28	-21.28	0.00	0: 0:23	0.000	19.	1.803	-2.565	34.090		
2032.	2.	.324	217.	214.	-21.84	-21.84	-0.01	0: 0:46	1.340	38.	1.803	-2.223	30.762		
2532.	6.	.361	243.	239.	-25.20	-25.20	-0.03	0: 1: 6	2.528	54.	1.803	-7.211	21.148		
3032.	266.	.378	254.	250.	-27.25	-27.25	-1.97	0: 1:24	3.832	70.	1.803	-8.921	17.821		
3532.	744.	.381	253.	251.	-27.36	-27.36	-3.55	0: 1:42	5.098	85.	1.803	-9.206	17.410		
4032.	1204.	.384	253.	253.	-27.63	-27.63	-3.44	0: 2: 0	6.357	99.	1.803	-9.578	16.785		
4532.	1666.	.387	253.	255.	-27.88	-27.88	-3.46	0: 2:18	7.612	113.	1.803	-9.939	16.175		
5032.	2126.	.390	253.	256.	-28.14	-28.14	-3.47	0: 2:36	8.862	127.	1.803	-10.293	15.572		
5532.	2587.	.394	253.	258.	-28.40	-28.40	-3.48	0: 2:54	10.116	141.	1.803	-10.629	14.990		
6032.	3046.	.397	253.	260.	-28.66	-28.66	-3.49	0: 3:11	11.346	154.	1.803	-10.962	14.408		
6532.	3503.	.400	253.	262.	-28.92	-28.92	-3.50	0: 3:29	12.580	167.	1.803	-11.292	13.829		
7032.	3963.	.404	253.	263.	-29.19	-29.19	-3.51	0: 3:46	13.819	179.	1.803	-11.611	13.248		
7532.	4420.	.407	254.	265.	-29.45	-29.45	-3.52	0: 4: 3	15.034	191.	1.803	-11.928	12.670		
8032.	4877.	.411	254.	267.	-29.72	-29.72	-3.53	0: 4:19	16.253	203.	1.803	-12.244	12.096		
8532.	5333.	.414	254.	269.	-29.99	-29.99	-7.09	0: 4:28	16.857	209.	1.803	-12.533	11.538		
8742.	5539.	.417	255.	271.	-30.32	-30.32	-3.20	0: 4:36	17.462	214.	1.803	-12.753	11.063		
9032.	5768.	.419	255.	272.	-30.43	-30.43	-3.57	0: 4:44	18.066	220.	1.803	-12.884	10.806		
9282.	5996.	.421	255.	273.	-30.56	-30.56	-3.54	0: 4:52	18.670	225.	1.803	-13.023	10.329		
9532.	6223.	.423	255.	274.	-30.70	-30.70	-3.55	0: 5: 1	19.272	231.	1.803	-13.057	10.411		
9782.	6450.	.424	255.	274.	-30.83	-30.83	-3.56	0: 5: 9	19.871	236.	1.803	-13.099	10.282		
10032.	6676.	.426	255.	275.	-30.96	-30.96	-3.57	0: 5:17	20.469	242.	1.803	-13.150	10.139		
10282.	6903.	.429	255.	276.	-31.10	-31.10	-3.58	0: 5:25	21.065	247.	1.803	-13.210	9.982		
10532.	7129.	.430	255.	277.	-31.23	-31.23	-3.59	0: 5:33	21.658	252.	1.803	-13.281	9.908		
10782.	7356.	.432	255.	278.	-31.36	-31.36	-3.60	0: 5:41	22.250	257.	1.803	-13.362	9.619		
11032.	7582.	.434	255.	279.	-31.49	-31.49	-3.63	0: 5:49	22.841	262.	1.803	-13.453	9.412		
11282.	7808.	.436	255.	280.	-31.63	-31.63	-3.61	0: 5:57	23.430	267.	1.803	-13.555	9.189		
11481.	7982.	.437	255.	281.	-31.78	-31.78	-2.80	0: 6: 4	24.018	272.	1.803	-13.659	8.958		

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

Table 7a. Summary of Optimum Results for a Climb-Cruise-Descent Profile.

	INITIAL CRUISE	FINAL CRUISE		INITIAL CRUISE	FINAL CRUISE
WEIGHT(LB)	131555.	131262.	TAS	482.	482.
COST(\$/NM)	2.313	2.310	IAS	313.34	313.03
ENERGY(FT)	41447.	41488.	GR SP KM	444.04	444.04
ALTITUDE	31186.	31230.	MACH NO	.82147	.82149
	FUEL USED(LB)	DISTANCE(N M),HR:MIN:SEC,		COST(), \$/NM	
CLIMB	4444.97	109.28	0:17:32	424.43	3.88
CRUISE	292.79	15.93	0: 2: 5	35.82	2.31
DESCEND	523.23	76.27	0:14:33	154.05	2.02
TOTAL	5260.97	201.09	0:34:11	614.30	3.05
LANDING WEIGHT =	130739.				
CRUISE AND OVERALL EFFICIENCY	19.853	26.163LB/NM			
COST (\$ OVER LAMDA) =	5.51NO OF ITERATIONS =	3			

Table 7b. Summary of Optimum Results for a Cruise-Descent Profile.

	INITIAL CRUISE	FINAL CRUISE		INITIAL CRUISE	FINAL CRUISE
WEIGHT(LB)	131807.	131413.	TAS	471.	471.
COST(\$/NM)	2.194	2.191	IAS	278.78	278.20
ENERGY(FT)	45938.	46612.	GR SP KM	446.50	446.39
ALTITUDE	36135.	36214.	MACH NO	.82081	.82068
	FUEL USED(LB)	DISTANCE(N M),HR:MIN:SEC,		COST(), \$/NM	
CLIMB	0.00	0.00	0: 0: 3	0.00	0.00
CRUISE	394.39	23.10	0: 3: 6	50.56	2.19
DESCEND	748.19	86.90	0:12:58	154.94	1.78
TOTAL	1142.58	110.00	0:16: 4	255.50	1.87
LANDING WEIGHT =	130664.				
CRUISE AND OVERALL EFFICIENCY	17.372	10.397LB/NM			

ORIGINAL PAGE IS
OF POOR QUALITY

one run against another. An example of a two-part trajectory (cruise-descent) is shown in Table 7b. Tables 7 are output for any value of the printout flag IPRINT.

Table 8 is an example of the table of optimum trajectory variables which may also be written to a data set for use in additional plotting routines or for the program TRAGEN if IGRAF \geq 1. Note that this summary shows GAMMA, which is the flight path angle with respect to the air mass, rather than the FPTH shown in Tables 6a and 6b. FPTH is the flight path angle with respect to the ground.

Table 9 is a summary of the cruise performance which is printed at the end of each run in which a cruise section is produced. Table 9 shows the steady cruise-climb conditions every 100 n.mi. to cover the expected range of the flight. Note that as fuel is burned off, the optimum altitude rises, and the optimum power setting and cost per n.mi. change. If cruise altitude is fixed, this will be indicated in the various cruise tables. Table 9 may also be written to Unit 11.

Table 10 is an example of the step climb portion of the profile calculated when step climb is included in the optimization.

Figures 1a-1b are examples of plots produced when IGRAF $>$ 1.

Unit 11 Output data set (optional)

The variables which describe the optimum vertical profile followed by the aircraft are written as output on Unit 11 when IGRAF \geq 1. Printer plots are also generated when IGRAF is greater than 1.

The output data are used for two purposes: (a) they serve as input to plotting routines so that a more convenient record of the data than tabular listings can be obtained, and (b) they serve as input of points on the nominal optimum trajectory for the program TRAGEN described in Ref. 1.

Table 8. Climb Trajectory Details. (IPRINT < 3).

ENERGY	ALTITUDE	MACH	AIRSPEED	OPTIMUM TRAJECTORY VARIABLES - CLIMB					TIME	DISTANCE	HDOT	VTASK
				GAMMA	FUEL/2	EPR	UWA					
1903.84	0.00	.31	207.46	0.00	0.00	1.80	0.00	0.00	0.00	0.00	0.00	210.00
2172.18	2.31	.33	221.48	.08	29.23	1.80	0.00	4.88	.29	.29	29.38	224.57
2672.18	6.39	.37	245.49	.07	79.94	1.79	0.00	13.25	.83	.83	30.78	249.69
3172.18	500.47	.37	245.74	8.07	130.69	1.80	0.00	21.74	1.41	1.41	3495.66	248.19
3672.18	943.33	.38	247.46	7.46	181.33	1.80	0.00	30.31	1.99	1.99	3253.75	248.28
4172.18	1427.76	.38	249.08	7.36	231.90	1.80	0.00	38.97	2.58	2.58	3231.68	248.27
4672.18	1891.52	.38	250.72	7.23	282.38	1.81	0.00	47.71	3.18	3.18	3195.69	248.27
5172.18	2354.77	.38	252.37	7.10	332.78	1.82	0.00	56.54	3.79	3.79	3161.34	248.27
5672.18	2817.47	.39	254.04	6.98	383.11	1.82	0.00	65.45	4.41	4.41	3126.66	248.27
6172.18	3279.64	.39	255.72	6.85	433.38	1.83	0.00	74.45	5.05	5.05	3091.73	248.26
6672.18	3741.25	.40	257.41	6.73	483.59	1.83	0.00	83.55	5.69	5.69	3056.54	248.26
7172.18	4202.31	.40	259.11	6.61	533.76	1.84	0.00	92.74	6.35	6.35	3021.11	248.26
7672.18	4662.79	.40	260.83	6.49	583.87	1.84	0.00	102.03	7.01	7.01	2986.14	248.25
8172.18	5122.71	.40	262.56	6.37	633.91	1.85	0.00	111.42	7.69	7.69	2952.90	248.25
8672.18	5582.04	.41	264.31	6.26	683.88	1.85	0.00	120.89	8.38	8.38	2919.37	248.25
9172.18	6040.78	.41	266.07	6.14	733.79	1.86	0.00	130.47	9.08	9.08	2885.56	248.24
9672.18	6498.92	.41	267.84	6.03	783.65	1.86	0.00	140.15	9.79	9.79	2851.47	248.24
10172.18	6956.46	.42	269.62	5.92	833.47	1.87	0.00	149.93	10.52	10.52	2817.12	248.24
10672.18	7413.38	.42	271.42	5.81	883.25	1.87	0.00	159.82	11.26	11.26	2782.51	248.24
11172.18	7869.68	.42	273.24	5.70	933.02	1.88	0.00	169.82	12.01	12.01	2748.65	248.23
11672.18	8325.34	.43	275.07	5.59	982.76	1.88	0.00	179.94	12.78	12.78	2714.56	248.23
12172.18	8780.37	.43	276.91	5.48	1032.50	1.89	0.00	190.16	13.56	13.56	2680.20	248.23
12672.18	9234.75	.44	278.76	5.37	1082.24	1.89	0.00	200.51	14.35	14.35	2645.59	248.22
13172.18	9688.47	.44	280.63	5.27	1132.00	1.90	0.00	210.98	15.16	15.16	2610.76	248.22
13672.18	10141.53	.44	282.52	5.16	1181.79	1.90	0.00	221.57	15.98	15.98	2575.25	248.22

Table 9. Cruise Portion Summary of Optimum Profile. (IPRINT < 3).

CRUISE PERFORMANCE TABLE									
CRUISE DIST NM	TIME HRS:MM:SS	WEIGHT LB	ENERGY BT	ALTITUDE FT	MACH NO	KTAS	CRS SPEED KNOT	LAMDA S/NM	PWR SETG EPR
0.00	0: 0: 0	131855.	46879.	36285.	.0200	470.25	463.99	2.188	1.933
100.00	0:13:27	129375.	46379.	36597.	.0200	470.25	458.07	2.174	1.935
200.00	0:26:54	127717.	46047.	36866.	.0200	470.25	456.05	2.161	1.936
300.00	0:40:21	126060.	45916.	37135.	.0200	470.25	456.12	2.149	1.936

ORIGINAL PAGE IS
OF POOR QUALITY

Table 10. Step Climb Portion of the Optimum Profile. (IPRINT < 3)

ORIGINAL PAGE IS
OF POOR QUALITY

CLIMB BEGINS AFTER		200.117 N.M. OF CRUISE		STEP CLIMB VARIABLES		EPR	V0	TIME	DISTANCE	EDOT	WEIGHT	H001
0	ENERGY ALTITUDE	MACH	AIRSPED	GAMMA	FUEL/2							
40004.007	33000.000	.724	422.197	0.000	0.000	2.277	422.116	0.000	0.000	21.928	94063.40	0.000
41344.287	33500.000	.725	421.110	1.923	41.250	2.233	421.037	20.959	2.453	20.905	94022.23	1431.344
41904.257	34000.000	.725	420.027	1.030	83.743	2.240	419.001	42.954	3.020	19.001	93979.74	1363.962
42264.200	34500.000	.725	410.954	1.757	127.577	2.240	410.740	66.004	7.712	18.023	93935.91	1297.015
42724.110	35000.000	.725	417.872	1.463	172.905	2.249	417.677	90.517	10.340	17.024	93890.50	1227.043
43104.042	35500.000	.725	416.700	1.561	220.437	2.257	416.012	116.615	13.571	16.410	93842.05	1149.533
43644.045	36000.000	.724	415.701	1.457	270.320	2.254	415.547	144.647	16.000	15.176	93795.16	1070.170
44120.324	36500.000	.724	415.273	1.201	325.910	2.257	415.139	174.554	20.407	13.030	93757.37	940.109
44623.145	37000.000	.724	415.132	1.143	384.075	2.257	415.020	212.334	24.610	12.430	93714.61	830.493

The output has up to six binary records of the form:

Record 1: WORD, NWORD

WORD may be: CLIMB, CRUISE, OR DESCEND.

NWORD is the number of points stored for the specified flight segment.

Record 2: An NWORD by 12 matrix. For example, for Climb,

Record 2 contains for JCLIMB = 1,...,NWORD:

CGRAF(JCLIMB,1)	=	E	Specific energy - ft
CGRAF(JCLIMB,2)	=	J	altitude - ft
CGRAF(JCLIMB,3)	=	MACH	Mach
CGRAF(JCLIMB,4)	=	VTASK	true airspeed - kt
CGRAF(JCLIMB,5)	=	GAMMA	flight path angle - deg
CGRAF(JCLIMB,6)	=	FUEL/2	fuel burned - lb
CGRAF(JCLIMB,7)	=	EPR	EPR setting
CGRAF(JCLIMB,8)	=	0	blank
CGRAF(JCLIMB,9)	=	TIME	time - sec
CGRAF(JCLIMB,10)	=	DIST	range traveled - n. mi.
CGRAF(JCLIMB,11)	=	HDOT	vertical rate - ft/min
CGRAF(JCLIMB,12)	=	VIASK	indicated airspeed - kt

The same variables are stored in DGRAF (JDESCN,J) for the descent portion, and SGRAF for the combined cruise and step climb portions.

ORIGINAL PAGE IS
OF POOR QUALITY

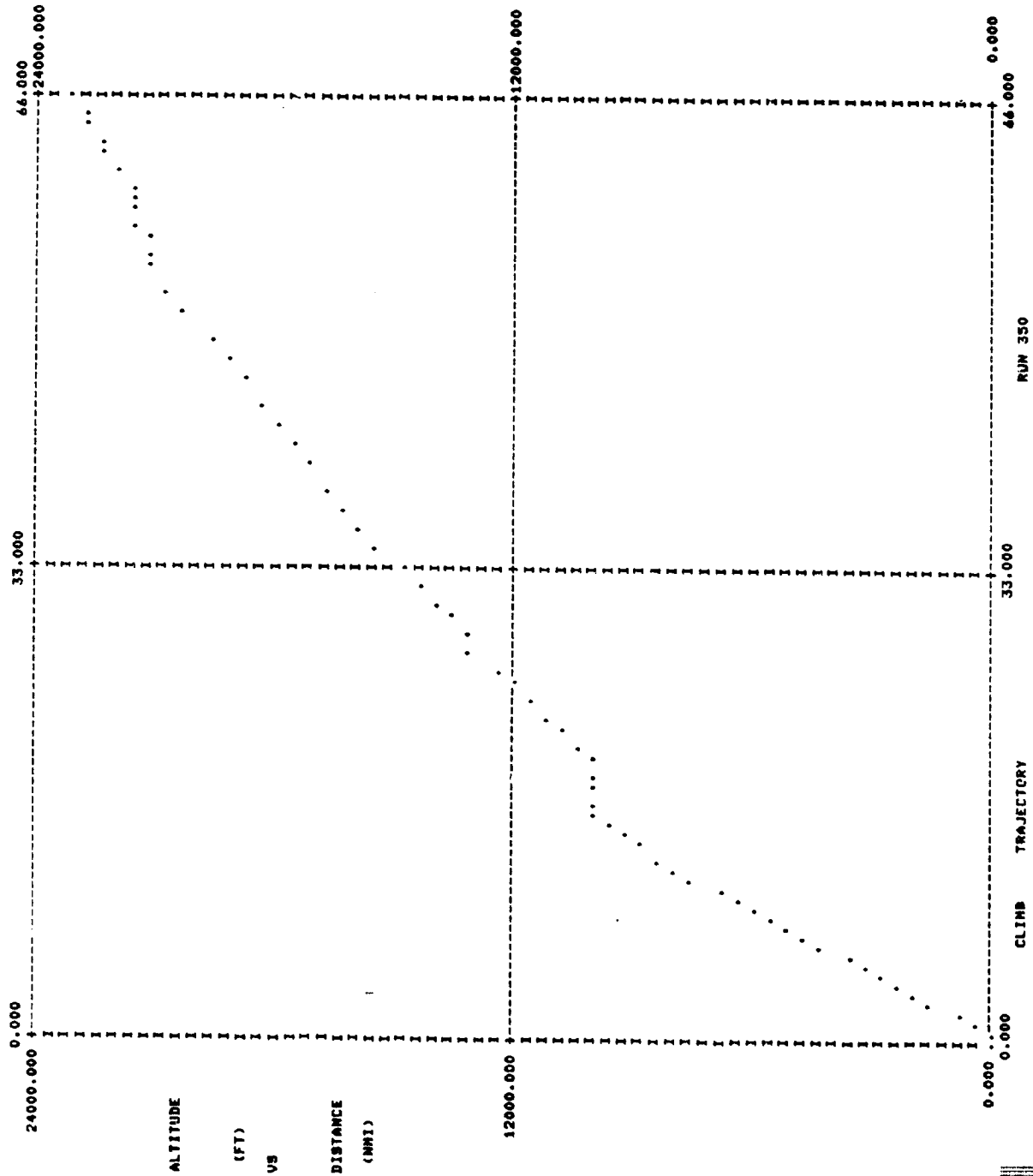


Figure 1a. Printer Plot Example. Altitude vs Range

ORIGINAL PRINT IS
OF POOR QUALITY

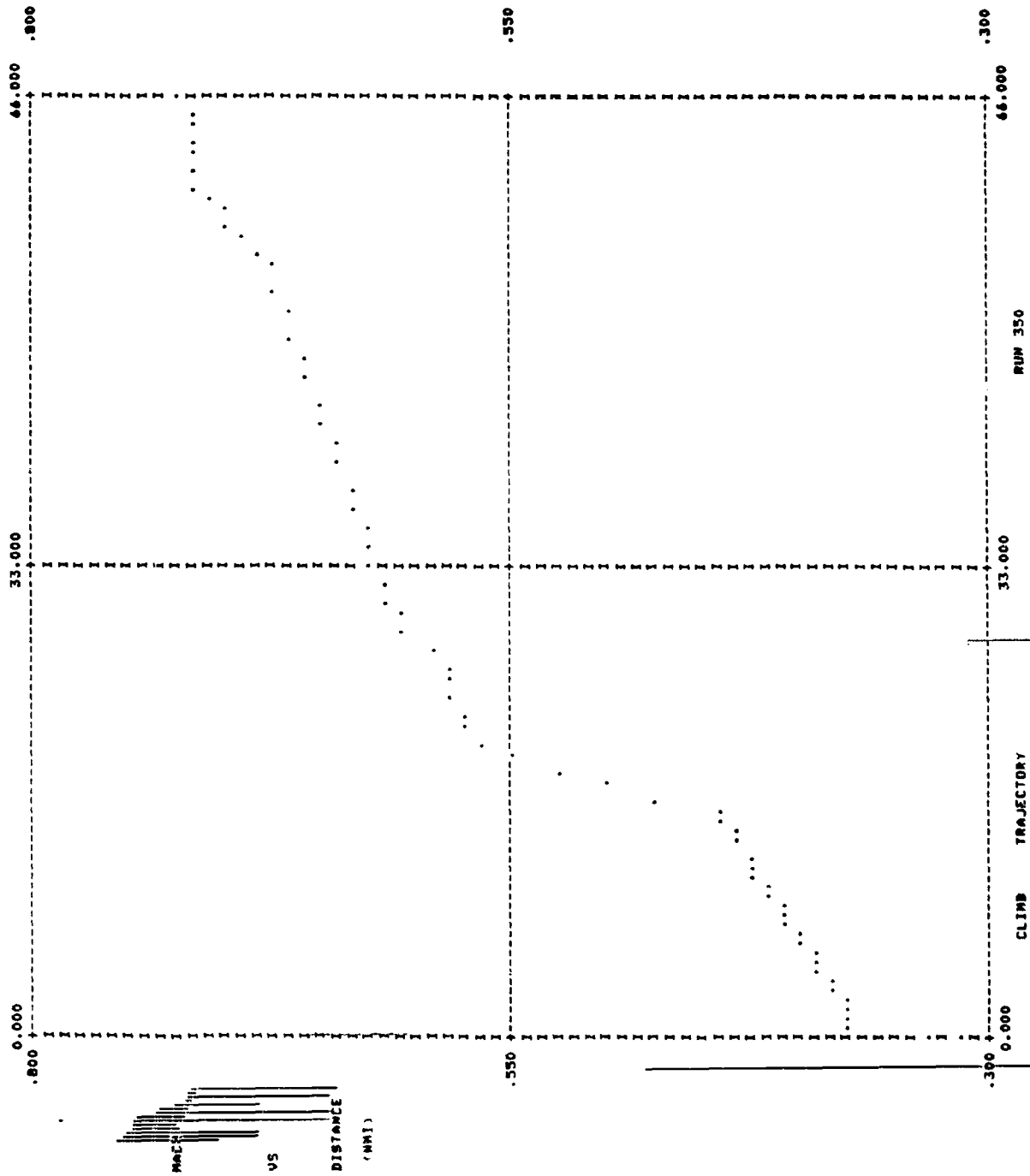


Figure 1b. Printer Plot Example. Mach Number vs Range.

IV

PROGRAM ORGANIZATION

This section gives a brief overview of the process used in OPTIM to generate an optimum vertical profile for aircraft operating between two fixed points. The overview is in the form of steps and program flow charts. Then a brief description is given of each of OPTIM's 62 subroutines and 19 functions. Charts showing the interrelations between these subroutines are also presented.

The technical details upon which the program is based are presented in Appendix A. A more detailed description of the subroutines is presented in Appendix B. More details concerning the program's origin and concepts upon which it is based can be found in Refs. 2-6.

OPTIM has been configured with a short main program which calls the input subroutine, ALLIN, and the major control subroutine, OPTM56. OPTM56 then follows one of two paths depending on the input parameters. It may synthesize a fixed-range, two-or three-part trajectory. Or, it may call OPTTOA to determine a trajectory meeting a fixed time-of-arrival constraint.

The fixed time-of-arrival is accomplished by iterating on the cost of time TC between passes of the program. In other words, an outer loop is used to iterate on the cost-of-time coefficient TC so that a fixed time-of-arrival is achieved. The inner loop determines the optimum vertical profile with a fixed-cost-of-time set by the outer loop and a fixed cost-of-fuel FC which is input.

Figure 2 shows a flow chart of the steps followed by OPTIM to synthesize a fixed-range, two- or three-part trajectory consisting of climb, cruise, and descent profiles. For a case where fixed-time-of-arrival is desired (ICTAB=2), Fig. 2 represents the inner loop of the program. (The outer loop is discussed later.) For the input flag ICTAB set at 0 or 1, this is the normal program flow. The steps followed are:

1. Read all input and place it in COMMONs/INPUT/ and /CRUISE/.
(The latter is filled by input only if an old cruise table is being read.)
2. Generate the cruise table if it has not been read in.
3. Test on trajectory type (input variable ICALT). If this is a two-part trajectory, go to (5a).
4. Set $P_{\min} = 1.00$ or 0.0 based on flags IVPI and ICTAB. Use this and other quantities to generate an optimum trajectory of range R_{\max} . If R is less than R_{\max} , go to (5). If R is within ϵ of R_{\max} , then the desired trajectory has been generated. If R is greater than R_{\max} , then the trajectory achieves optimum cruise altitude. If so, compute the cruise distance $d_c = R - (d_{\text{up}} + d_{\text{dn}})$, and compute the final cruise weight. Next, use this updated cruise weight to recompute a refined descent trajectory. Use this refinement to complete the three-part trajectory.
- 4a. For a two-part trajectory, use λ_{opt} and input quantities to generate the optimum descent profile. Compute the cruise distance $d_c = R - d_{\text{dn}}$, and compute the final cruise weight. Use this updated cruise weight to recompute a refined descent trajectory. Use this refinement to recompute the two-part trajectory. (Note, if initial altitude is not specified, the starting optimum cruise altitude and λ_{opt} are chosen based on the input initial cruise weight. If initial cruise altitude is specified, this is fixed, and λ_{opt} is obtained from the cruise table corresponding to the initial altitude and weight.)
5. Compute P_{\max} that causes cruise altitude just above 10000 ft. (See Appendix A). Use the input quantities and P_{\max} to generate the optimum trajectory of range R_{\min} . Compare input range R with R_{\min} . If R is less than R_{\min} , then no trajectory is computed. If R is within ϵ of R_{\min} , then the desired trajectory has been synthesized ($\epsilon = 5$ n.mi.). If R is greater than R_{\min} , then go on to (6).
6. For $R_{\min} < R < R_{\max}$, use (P_{\max}, R_{\min}) and (P_{\min}, R_{\max}) as the starting points to compute subsequent values of P . Iterate on P until the desired range is achieved. (Again, refer to Appendix A for more details.)

In steps (2), (3), and (4) of Fig. 2, an optimum trajectory is generated based on the parameter P . Computing the points on an optimum profile and evaluating the results for a fixed value of λ consists of nine steps which are presented in flow chart form in Fig. 3. The input quantities are P , the initial weight W_1 , and the initial and final values of specific energy (E_1, E_f) (which are computed from initial and final altitude and airspeed (h_1, V_1), (h_f, V_f)). The steps follow the analytical expressions developed in Appendix A, and they are:

1. The climb fuel burned is estimated based on an empirical equation which is a function of initial and final energy, the cost parameters C_f and C_t , and the initial weight. The climb fuel estimate is subtracted from the initial weight to obtain an estimate of the initial cruise weight W_{c1} . Based on W_{c1} , the optimum cruise cost λ_{opt} is obtained from the cruise table. This λ_{opt} term is multiplied by a percentage P to obtain the cost parameter λ_1 used for the optimization of the climb profile. This modified cost term and W_{c1} are used to interpolate in the cruise table to obtain the estimate of the initial cruise energy E_{c1} . If a time-of-arrival or cruise altitude are fixed, then $P = 0$, and $\lambda = \lambda_{opt}$.
2. The optimum climb trajectory is generated by stepping along in energy from E_1 to E_{c1} , in steps of ΔE . At each point, the Hamiltonian

$$H_{up} = V, \pi \left\{ \frac{C_f \dot{E} + C_t - \lambda_1 V_g}{\dot{E}} \right\}; \dot{E} > 0,$$

is minimized by choosing the best value of airspeed V (IVPI = 0) or airspeed and EPR π (IVPI = 1). Constraints limit the range of airspeed and thrust over which the search takes place. Energy and altitude are monotonically increasing. Based on the change in energy and airspeed, the approximate changes in altitude, flight path angle γ , time t , fuel burned F , and distance traveled X are computed. This process is stopped when energy E_{c1} is reached. If the weights at top of climb is more than 200 lb off the estimate W_{c1} , this step is represented with improved values of λ_1 .

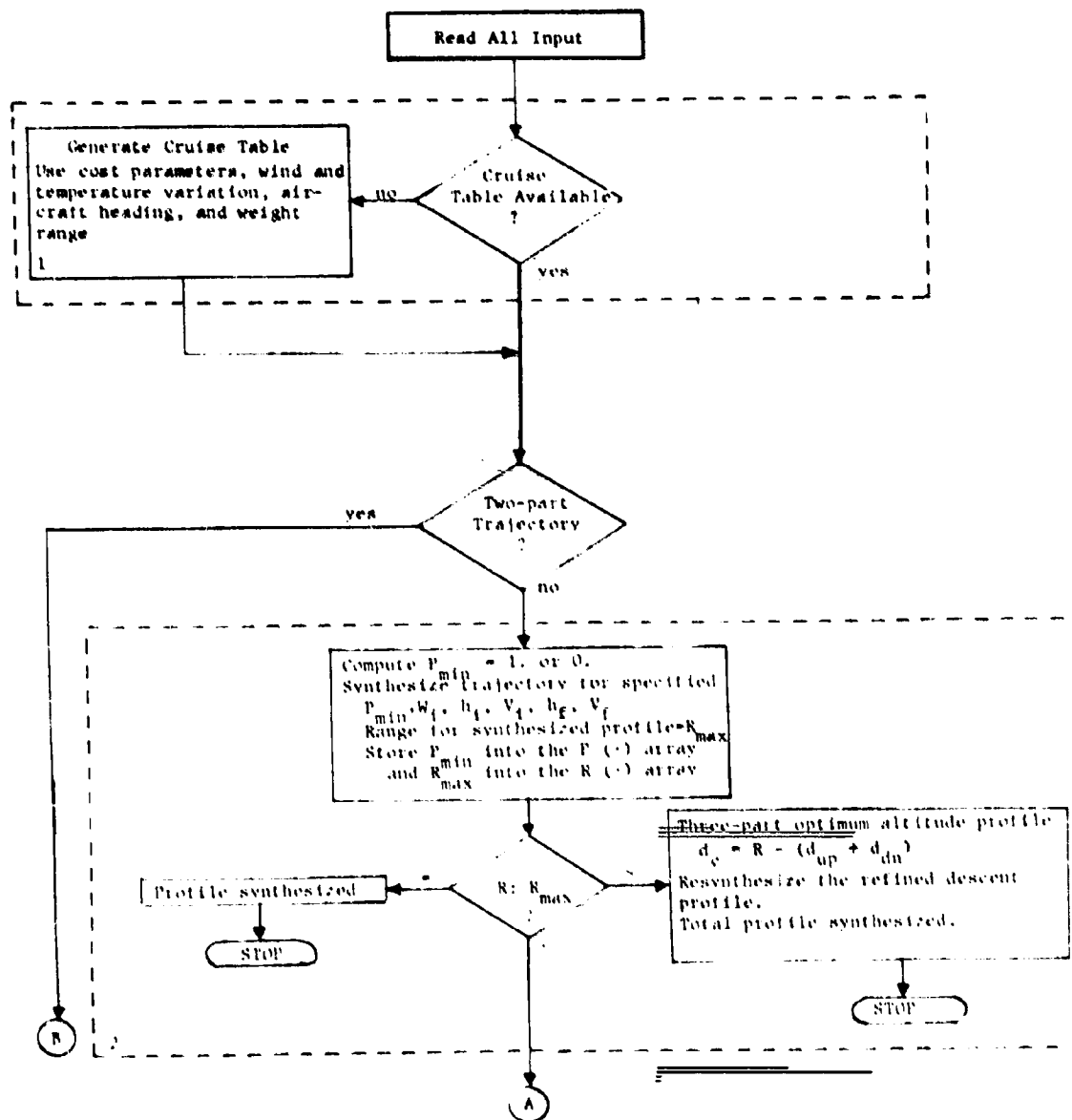


Figure 2. Macro Flow Chart for Synthesizing a Fixed-Range, Three-Part Optimum Profile

ORIGINAL PAGE IS
OF POOR QUALITY

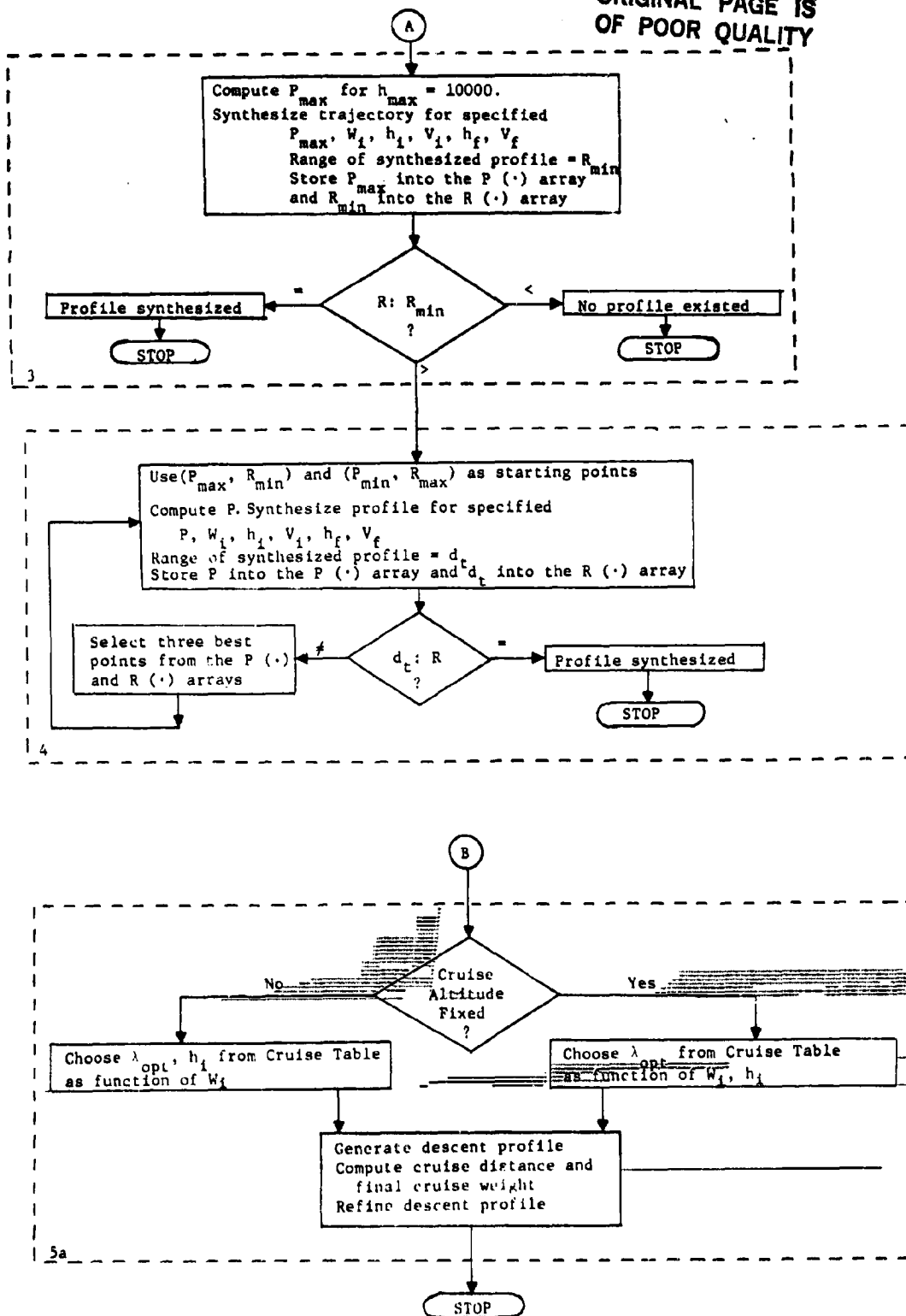
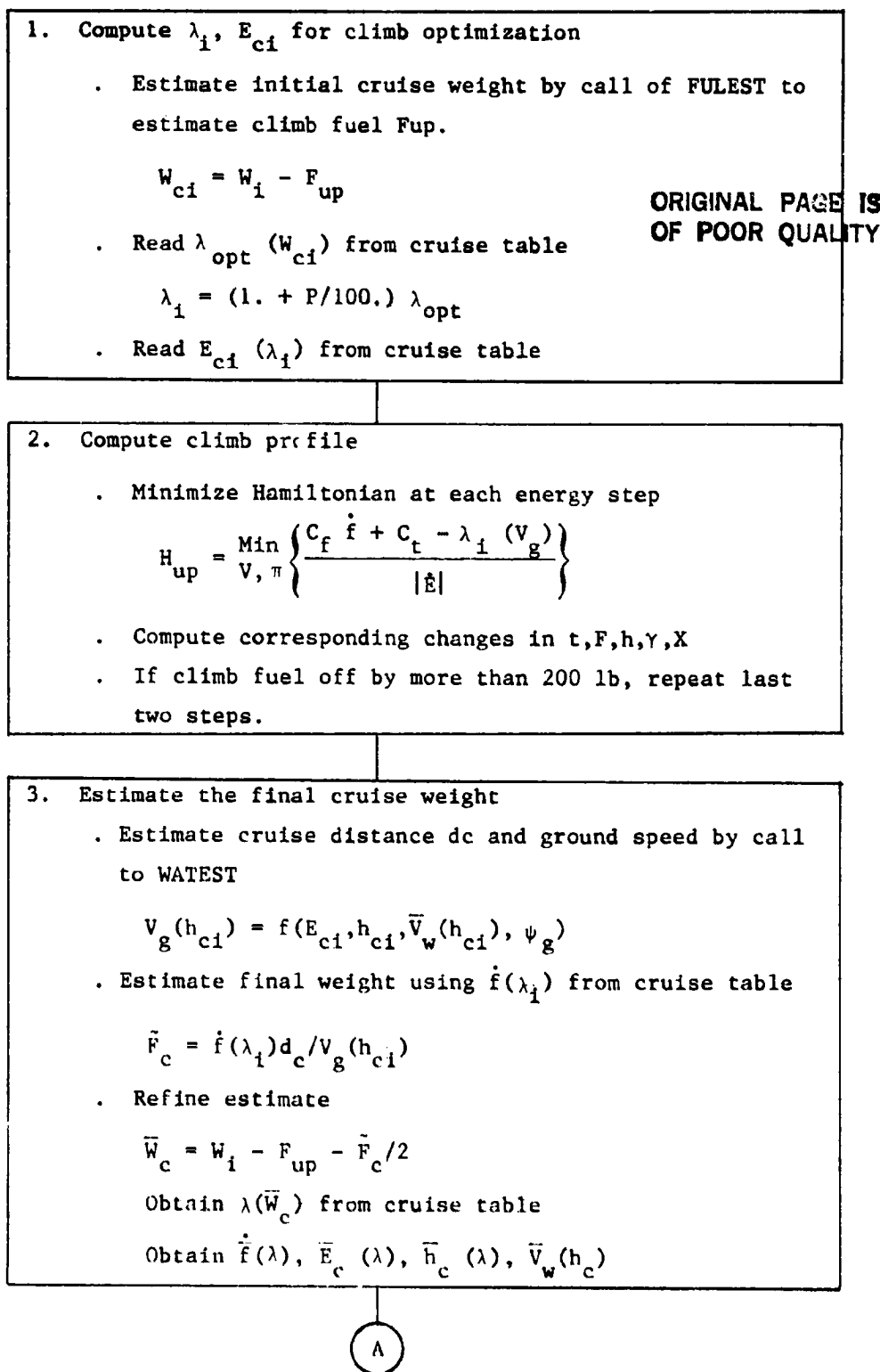


Figure 2. Continued.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 3. Nine-step process to generate an optimum three-part trajectory

$$\text{Compute } \bar{V}_g(\bar{E}_c, \bar{h}_c, \bar{V}_w)$$

$$F_c = \bar{f} d_c / \bar{V}_g$$

- Compute final weight estimate

$$W_{cf} = W_i - F_{up} - F_c$$

4. Compute cruise cost and energy for beginning of descent

- Obtain $\lambda_{cf}(W_{cf})$ from cruise table
- $\lambda_f = (1. + P/100.)\lambda_{cf}$
- Obtain $E_{cf}(\lambda_f)$ from cruise table

5. Estimate landing weight from WATEST estimate of descent fuel F_{dn} :

$$W_f = W_{cf} - F_{dn}$$

6. Compute initial descent profile

- Minimize Hamiltonian at each energy step

$$H_{dn} = \min_{V, n} \left\{ \frac{C_f f + C_t - \lambda_f V_g}{|E|} \right\}$$

- Compute corresponding changes in t, F, h, γ, X

7. Refine final cruise and landing conditions

- Recompute cruise distance

$$d_c = (H_{up} + H_{dn}) / (d\lambda/dE)$$

- Repeat Step (3) to compute F_c

$$W_{cf} = W_i - (F_{up} + F_c)$$

$$W_f = W_i - (F_{up} + F_c + F_{dn})$$

8. Recompute refined descent profile

- Repeat Step (6)
- Repeat Step (7) for improved cruise estimates

9. Tabulate results of Steps (1) - (8)

3. The final cruise weight W_{cf} is next estimated. For the climb-descent type of trajectory (IVPI = 1; see Appendix A) no fuel is burned in cruise, W_{cf} is set equal to W_{ci} , and the program proceeds to Step (4). Otherwise, the program uses empirical equations to compute cruise distance d_c . Then, with initial cruise cost λ_i , fuel burn rate, and estimated ground speed V_g , it computes an initial estimated fuel burned during cruise. This value is used to estimate the aircraft weight \bar{W}_c half-way through cruise. A new cruise cost $\bar{\lambda}$ and fuel burn rate \bar{f} are obtained from the cruise table. These values are used to refine the estimate F_c of the fuel burned and the final cruise weight W_{cf} .
4. The estimate W_{cf} is used to obtain from the cruise table the final optimum cruise values of cost λ_{cf} . The value of λ_f used for descent optimization is found by multiplying λ_{cf} by the percentage P. The final value of cruise energy E_{cf} used for descent optimization is found by interpolating from the cruise tables using W_{cf} and λ_f .
5. The landing weight is estimated using empirical equations.
6. Similar to Step (2), the optimum descent trajectory is generated by stepping along in energy (backwards in time) from E_f to E_{cf} , in steps of ΔE . At each point, the Hamiltonian

$$H_{dn} = \frac{1}{V, \pi} \left\{ C_f \dot{f} + C_t - \lambda_f V_g \right\} / |E| ; \dot{E} < 0 ,$$

is minimized. Again, changes in altitude, flight path angle, time, fuel burned, and distance traveled during descent are computed.

7. The cruise distance is refined. For aircraft reaching optimum cruise altitude ($P = 0$), cruise distance d_c is R -

$d_{up} - d_{dn}$. Otherwise, a formula using the final values of H_{up} , H_{dn} , and the slope $d\psi/dE$ are used to compute d_c . With this more accurate distance, improved values of W_{cf} and W_f are computed.

8. The descent profile is recomputed based on the improved value of W_f . This produces a better value of descent distance d_{dn} which in turn produces a better value of cruise distance d_c .
9. Based on Steps (1)-(8), a table (Table 7a) is generated which summarizes the distances traveled, end conditions, fuel burned, and costs of each segment of the trajectory plus the trajectory as a whole.

The above steps and flow diagrams of Figs. 2 and 3 are brief summaries of the process taken by the OPTIM program to implement the vertical profile optimization techniques outlined in Appendix A. The reader who is interested in more program details is referred to Ref. 3. Further understanding will come from use of the program and study of the individual subroutines.

Two other variations to the basic process of generating optimum vertical profiles exist within the structure of OPTIM. The first is the ability to simulate a step climb during cruise. This climb is currently 4000 ft from one fixed cruise altitude to another (e.g., 33000 ft to 37000 ft). For this option, the step climb is computed based on use of maximum climb thrust, and the speed is ramped from the optimum at the lower altitude, to the optimum at the higher altitude. The climb computation consists of eight 500 ft steps, with the aircraft trimmed so that flight path angle rate for each step is zero. Then, optimum cruise and descent are computed from the higher cruise altitude. The program is set up to compute the optimum-point to begin the step climb if this option is chosen.

The second variation is to constrain the rate of descent at the top of the descent portion of the profile. This option might be used to account for cabin

pressurization constraints. Currently, OPTIM has the ability to constrain the rate at 500 ft/min down to 28000 ft. Then, an optimum descent profile is computed from this point downward, where λ_f is based on the aircraft weight at 28000 ft.

The capability to generate a vertical optimum profile with fixed range and fixed time-of-arrival is governed by the outer loop logic shown in Fig. 4. The program begins (as also shown in Fig. 2) by reading in control flags and other trajectory characteristics data. If the control flag ICTAB is set to 0 or 1, OPTIM generates a minimum cost optimum profile based on the input cost-of-fuel FC and cost-of-time TC. This is the normal mode of operation.

If the flag ICTAB is set to 2, the program will iterate on the value of cost of time TC until the time-of-arrival T_f is within some tolerance of the desired time-of-arrival T_{end} (input TEND). The logic for this iteration scheme is shown as Blocks 2, 3, and 4 in Fig. 4.

The first step is to set TC to zero and generate an optimum vertical profile. This profile will correspond to a minimum fuel path. The logic to generate this profile is essentially the same as was indicated in Figs. 2 and 3. The final time T_{fo} is recorded. If T_{fo} is greater than T_{end} , the initial flight profile took too much time. Therefore, time must be penalized with positive cost TC. The program then uses logic indicated in Block 3a. If T_{fo} is less than T_{end} , the initial flight profile was too fast. Then, time must be penalized with negative TC. The program then uses logic indicated in Block 3b.

For positive TC, the program generates two more optimum profiles corresponding to costs of time of \$300/hr and \$600/hr. (If \$600/hr produces too slow a profile, the program uses \$900/hr, etc.) From each of these, the times-of-arrival T_{f1} and T_{f2} are recorded. Then, the program proceeds to Block 4 to solve for a TC which will yield the desired T_f .

For negative TC, the program next solves for the minimum cruise value of TC, or

$$TC_{min} = -\dot{f}_{min} FC.$$

ORIGINAL PAGE IS
OF POOR QUALITY

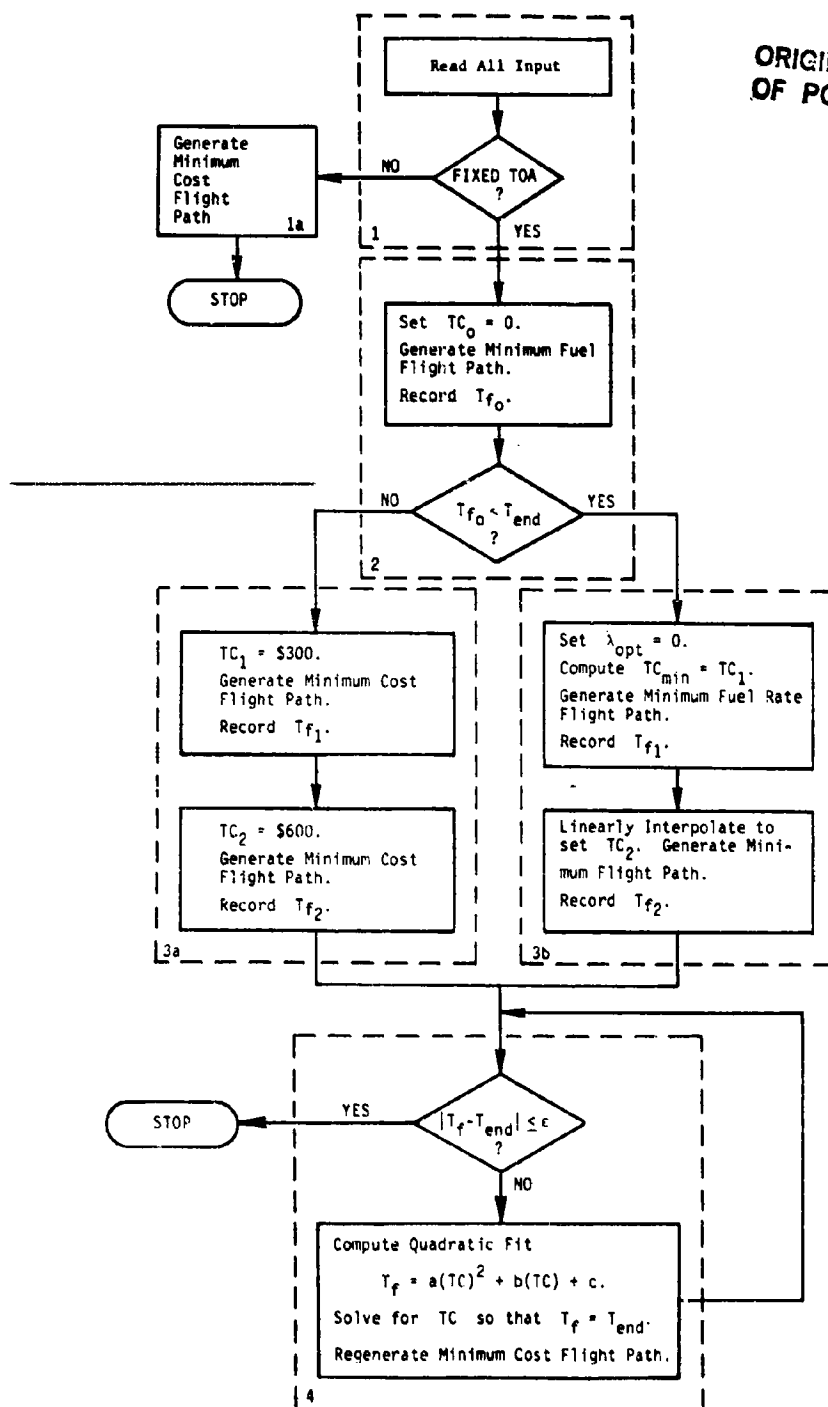


Figure 4. Macro Flow Chart for Synthesizing a Fixed Range,
Fixed Time-of-Arrival Optimum Profile

FC is the input cost of fuel, and \dot{f}_{\min} is the minimum cruise fuel rate at a particular cruise altitude. The aircraft should not go slower in cruise than the cruise speed corresponding to \dot{f}_{\min} . Thus, TC_{\min} represents the boundary on negative values of TC. For this value of TC, the variable λ_{opt} (see Fig. 1 and Appendix A) is zero. A minimum fuel rate profile is then generated using TC_{\min} , and the corresponding time of flight T_{f1} is recorded. (If T_{f1} is less than T_{end} , the difference should be made up in a holding pattern at the end of cruise.) For T_{f1} greater than T_{end} , the next value of TC is set by linear interpolation between 0 and TC_{\min} to attempt to produce a T_{f2} equal to T_{end} .

If T_{f2} equals T_{end} , the program is finished. If not, T_{f2} is recorded, and the program proceeds to Block 4.

In Block 4, three values of time-of-arrival T_f (T_{f0} , T_{f1} , T_{f2}) are used with three corresponding values of cost-of-time TC (TC_1 , TC_2 , TC_3) to form the quadratic relationship

$$T_f = a(TC)^2 + b(TC) + C.$$

This equation is solved for TC so that $T_f = T_{\text{end}}$. Then the program is rerun with this new value of TC used to generate the optimum profile. The resulting T_f is compared to T_{end} . If it is within ϵ (10 sec), the program stops. If not, the new values of (TC, T_f) are used with previous values to recompute the quadratic relationship, and the process is repeated. This continues until the generated flight paths converge to the desired time-of-arrival.

OPTIM is programmed in FORTRAN, and it consists of the main executive program, sixty-two subroutines, nineteen functions, and three block data routines. These subroutines and functions are called to execute the steps depicted in Figs. 2-4. Brief explanations of the program and its subroutines are presented in Appendix B.

The program subroutines and functions can be grouped into four categories:

1. trajectory optimization and generation,
2. aerodynamic and propulsion models,
3. flight condition models, and
4. utility programs.

ORIGINAL PAGE IS
OF POOR QUALITY

Under Category 1, the routines are:

OPTM56	Serves as the driver program after all input is in.
OPTTOA	Serves as the outer-loop driver program when searching on TC to achieve the desired time-of-arrival.
ALLIN	Reads all input.
CLIMB	Generates and stores the climb profile.
CRUISR	Computes a cruise segment from the lower cruise table before a step climb.
CRUISX	Compute a cruise segment from a given starting weight proceeding for a given range.
CRZOP5	Generates the cruise table.
CTABLE	Interpolates in the cruise table for a set of parameters at a given weight.
DESCND	Generates and stores the descent <u>profile.</u>
DESPC	Modifies the final portion of the cruise and descent profile to provide for constrained descent.
DRAGC	Computes the drag force.
ESTCD	Calls appropriate routine to estimate the cruise range.
ESTCD2	Estimates cruise range for the tri-jet as a function of percent P of λ_{opt} .
ESTCD3	Estimates cruise range for the twin-jet as a function of cost of time, cost of fuel, and P.
ESTDF	Call appropriate routine to estimate the descent fuel.
ESTDF2	Estimates the descent fuel for the tri-jet as a function of P.
ESTDF3	Estimates the descent fuel for the twin-jet as a function of cruise and landing energy, cost of time, and cost of fuel.
ESTEP	Computes the value of the next energy step during climb and descent.

FBOUND	Generates the speed boundaries for the optimization search.
FCLIMB	Minimizes the climb and descent Hamiltonians with respect to speed.
FCLMB6	Controls the process of minimizing the Hamiltonian for a fixed energy level during climb.
FCOST	Minimizes the cruise cost for a fixed altitude and aircraft weight.
FDSCN6	Controls the process of minimizing the Hamiltonian for a fixed energy level during descent.
FDRAG	Computes the minimum drag airspeed.
FOPT	Generates the optimum cruise cost for a given cruise weight.
FTHRST	Minimizes the climb and descent Hamiltonian with respect to thrust.
FULEST	Calls the appropriate routine to estimate the climb fuel.
FULEST2	Estimates climb fuel for the tri-jet.
FULST3	Estimates climb level for the twin-jet.
PCCMP5	Computes the value of F used as an iteration parameter to compute a trajectory with the desired range.
PILINT	Generates the lower (EPR) limit for climb optimization, and the upper limit for descent optimization.
PRETBL	Prints pre-step-climb cruise performance table.
PRFTBL	Prints performance table, writes data on Unit 11, and calls for printer plots.
PROFIL	Controls computation of one optimum flight profile.
PRSUM	Prints cruise table summary. Calculates summed cruise time and distance tables.
STEP	Controls the addition of a step climb to the optimization.
STEPEN	Takes one energy step (climb or descent) using the minimized Hamiltonian.
STEPOPT	Calculates cost of combined cruise, step climb, cruise and descent.
STEPUP	Computes a step climb in altitude and Mach number.

VOPTRJ Computes the fuel used, distance traveled, time taken, total cost and cost/n.mi. for the climb, cruise, and descent profiles.

WATEST Estimates the landing weight, given conditions at top of climb.

WCLST Computes final climb, cruise and descent values for fuel, time and distance, where the value of λ is very close to the optimum value.

WLEFHV Interpolates the cruise table data to relate λ , cruise weight, energy, fuel flow rate, altitude, and ground speed.

Under Category 2, the routines are:

DATTRI Block data containing engine data for the tri-jet turbofan engine.

DATTWN Block data containing engine data for the twin-jet aircraft.

CDRAG Calls appropriate routine to compute the drag coefficient.

CDRAG2 Computes the drag coefficient for the tri-jet aircraft.

CDRAG3 Computes the drag coefficient for the twin-jet aircraft.

CLIFTT Calls appropriate routine to compute the lift coefficient.

CLIFT2 Computes the lift coefficient as a function of Mach number, altitude, and angle-of-attack for the tri-jet aircraft.

CLIFT3 Computes the lift coefficient for the twin-jet aircraft.

ENGEPR Calls appropriate routine to compute engine thrust and fuel flow rate.

ENGEPR2 Computes the engine thrust and fuel flow rate as functions of altitude, Mach number, temperature variations, and EPR setting for the tri-jet.

ENGEPR3 Computes the engine thrust and fuel flow rate for the twin-jet aircraft.

ENGIDL Computes thrust and fuel flow rate for idle throttle.

SPLMT Computes aircraft speed limits during climb or descent.

TRIM Computes the thrust and angle-of-attack for maintaining constant speed levels for a given altitude and cruise weight.

Under Category 3, the routines are:

ATLOW	Generates atmospheric density, pressure, temperature, and speed-of-sound as functions of altitude.
WIND	Computes the wind vector and its effect along the ground track of the aircraft.
WINDIN	Reads in data and sets up the wind profile as a function of altitude.

Under Category 4, the routines are:

BANNER	Writes the heading at the beginning of the run.
CHEKIN	Checks input quantities to be sure they are within reasonable ranges.
CONDAT	Block data containing values for most program constants.
DBLSRC	Performs a linear double table look-up.
FIAS	Converts indicated airspeed in feet/second to Mach number.
FIASM	Converts Mach number to indicated airspeed in knots.
FMIN	Minimizes a function by a Fibonacci search to within 1/144 of the search interval.
FMIN2	Minimizes a function by a Fibonacci search to within 1/34 of the search interval.
ICLOCK	Changes time in seconds into hours, minutes, and seconds.
JTRUNC	Truncates a monotonically decreasing series from an array of changing values.
LSQPOL	Obtains a polynomial based on a least-squares fit to a set of data.
MATINV	Inverts a matrix.
NICER	Generates boundaries for printer plots.
PAGE	Starts a new page of printout.
PICTUR	Generates printer plots.
POLYE1	Evaluates a polynomial for some fixed value of the independent variable.
POLY2	Evaluates a polynomial for some fixed value of the two independent variables.

PRTPLT	Sets up printer plots when IGRAF is greater than 1.
PRWT	Prints estimated conditions at top of climb.
SERCHI	Searches for a point in a monotonically increasing array.
SERCHD	Searches for a point in the monotonically decreasing array.
SGLSRC	Performs a linear table look-up.
TRACIT	Traces subroutine calling sequence in case of program error.
WRITE1	Writes out the trajectory summary table.

Three CDC subroutines are called: DATE, TIME, and STRACE. These are for local user convenience.

Figures 5a through 5i show how control passes through the chain of subroutines. This figure, when combined with the preceeding short subroutine descriptions, is a short guide to the total program organization. Further detail may be found in Appendix B.

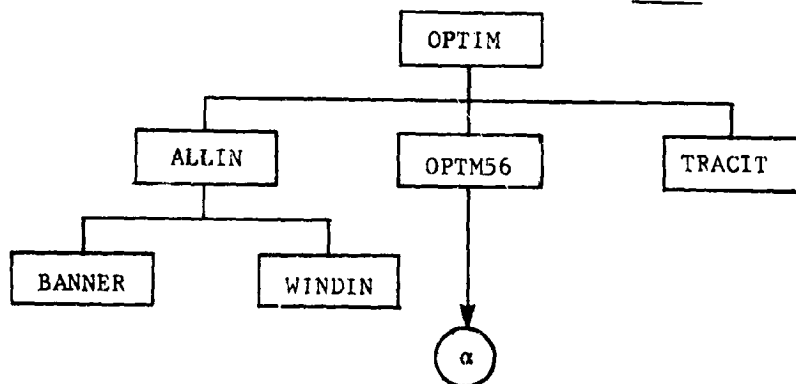


Figure 5a. Top Level of Program Flow.

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

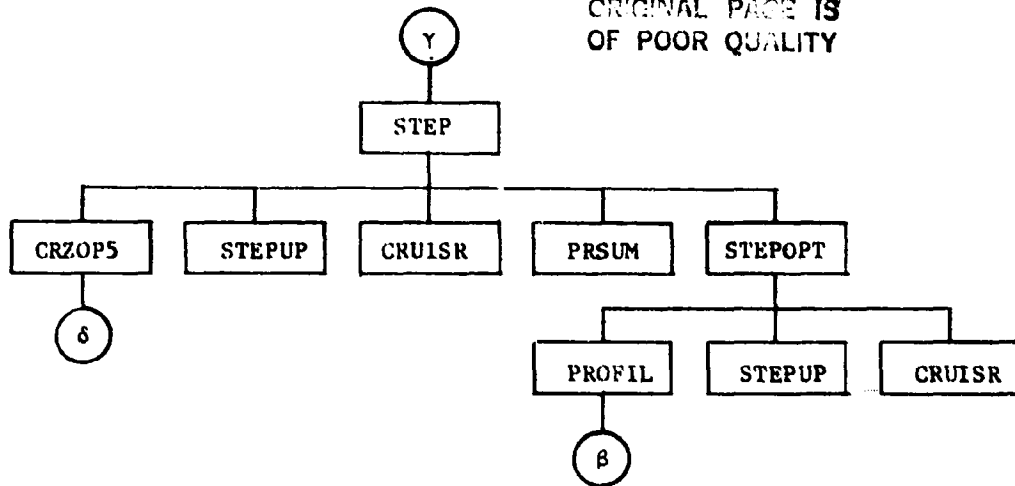


Figure 5d. Program Flow for Step Climb Optimization.

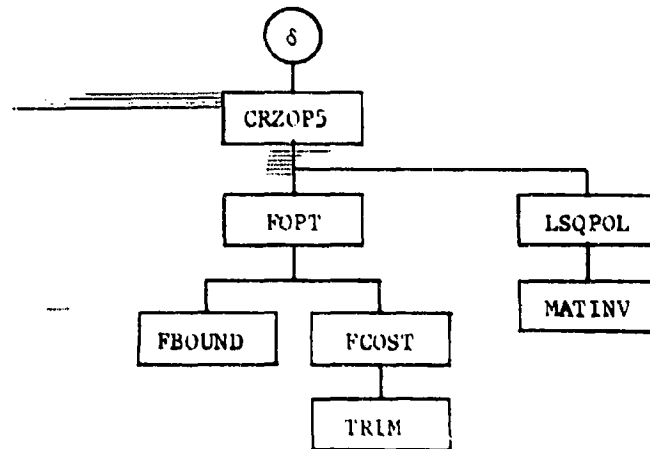


Figure 5e. Program Flow for Cruise Table Optimization.

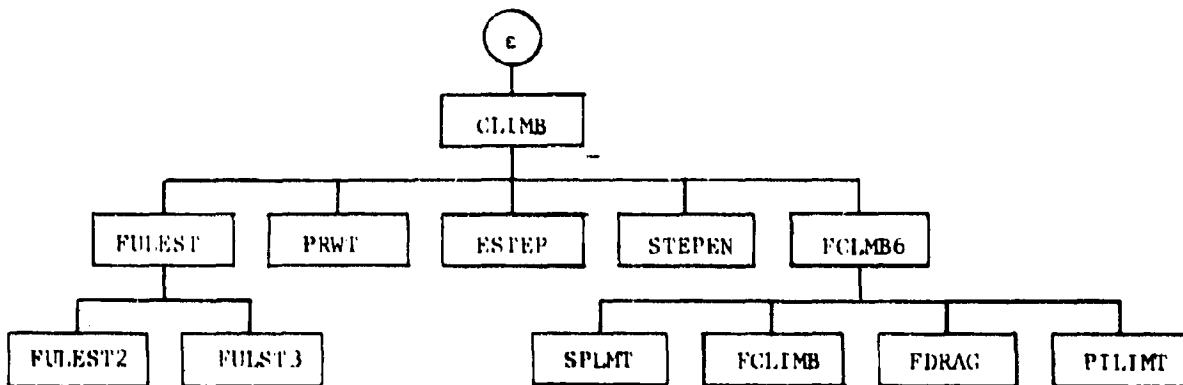


Figure 5f. Program Flow for Climb Optimization.

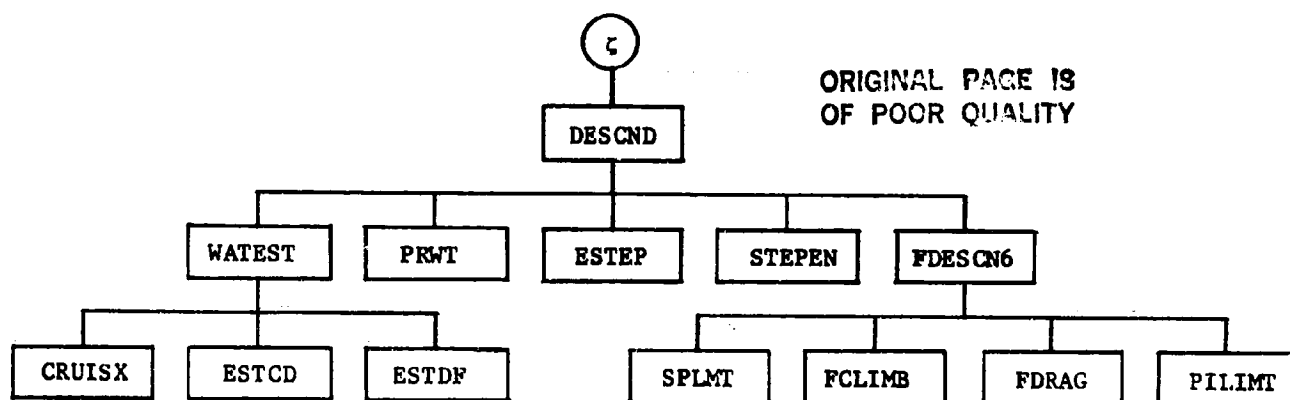


Figure 5g. Program Flow for Descent Optimization.

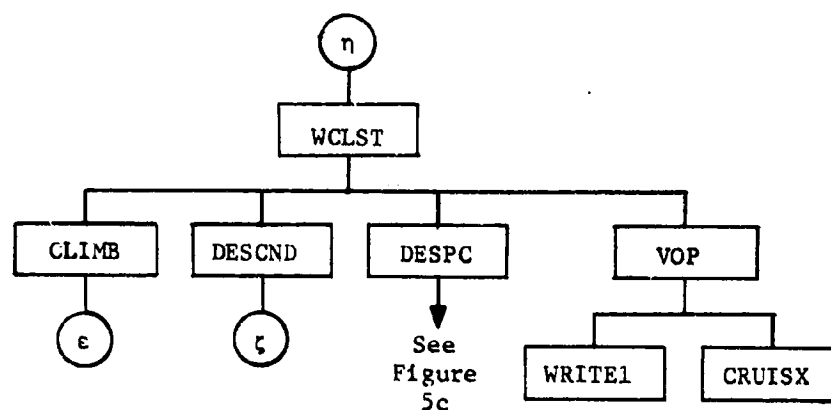


Figure 5h. Program Flow for Near-Optimum Case.

ATLOW	ENGEPR	FMIN2	POLY2
CDRAG	ENGEP2	FTHRST	SERCHD
CDRAG2	ENGEP3	ICLOCK	SERCHI
CDRAG(3)	ENGIDL	JTRUNC	SGLSRC
CTABLE	FIAS	PAGE	WIND
DBLSRC	FIASM	POLYE1	WLEFHV
DRAGC	FMIN		

Figure 5i. Utility Subprograms and Functions Called Throughout the Program.

APPENDIX A

TRAJECTORY OPTIMIZATION USING THE ENERGY STATE METHOD

The purpose of this Appendix is to summarize briefly the theoretical background and the numerical process used in the OPTIM program for computing the optimum vertical profile of a turbo-jet aircraft. More details are given in Refs. 2-6. Reference 7 presents the principles upon which trajectory optimization is based. In Refs. 2 and 3, Erzberger and Lee apply these principles using the energy state approximation to obtain a practical, efficient means of generating the optimum vertical profile. OPTIM is an extension of the original computer code developed by Erzberger and Lee and is based on their methods. Its application is explained in Refs. 4, 5, and 6.

In the following sections, the theory of trajectory optimization is first presented. Then, the application of this theory to minimizing the direct operating cost (DOC) of an aircraft traveling over a fixed range is outlined. This is followed by a discussion of the details of going from theoretical expressions to a practical computer code. The theoretical points are presented without proof, for conciseness. The reader wanting more detail should review the references.

Theoretical Principles

In Ref. 7., a description is given of the requirements for solving an optimization problem involving a continuous dynamic system with no terminal constraints but with fixed terminal time. This description is repeated here because it presents the basic principles which extend to the aircraft profile optimization problem.

ORIGINAL PAGE IS
OF POOR QUALITY

A system (the aircraft) is governed by the nonlinear differential equations

$$\begin{aligned} \dot{x} &= f(x, u, t) ; & x(t_0) &\text{ given;} \\ t_0 &\leq t \leq t_f ; \end{aligned} \quad (A.1)$$

where x is the n -dimensional state vector and u is the m -dimensional control vector. The cost function which is to be minimized is of the form

$$J = \phi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x, u, t) dt. \quad (A.2)$$

Here, ϕ is the terminal cost function, and L is the cost per unit time along the trajectory. The problem is to find the sequence of controls $u(t)$ that minimize J .

First, the system equations are adjoined to J with the multiplier vector $\lambda(t)$:

$$J = \phi(x(t_f), t_f) + \int_{t_0}^{t_f} \{L(x, u, t) + \lambda^T(t) \{f(x, u, t) - \dot{x}\}\} dt. \quad (A.3)$$

Then the Hamiltonian function is defined as

$$H(x, u, t) = L(x, u, t) + \lambda^T(t) f(x, u, t). \quad (A.4)$$

Equation (A.3) is integrated by parts to yield

$$\begin{aligned} J &= \phi(x(t_f), t_f) - \lambda^T(t_f) x(t_f) + \lambda^T(t_0) x(t_0) \\ &+ \int_{t_0}^{t_f} \{H(x, u, t) + \dot{\lambda}^T(t) x(t)\} dt. \end{aligned} \quad (A.5)$$

Next, the change in J due to variations in $u(t)$ and $x(t)$ is considered for fixed t_0 and t_f :

$$\delta J = \left\{ \left(\frac{\partial \phi}{\partial x} - \lambda^T \right) \delta x \right\}_{t=t_f} + (\lambda^T \delta x)_{t=t_0} + \int_{t_0}^{t_f} \left\{ \left(\frac{\partial H}{\partial x} + \dot{\lambda}^T \right) \delta x + \frac{\partial H}{\partial u} \delta u \right\} dt. \quad (A.6)$$

The elements of $\lambda(t)$ are chosen to cause the coefficients of δx in Eq. (A.6) to vanish under the integral and at t_f :

$$\dot{\lambda}^T = - \frac{\partial H}{\partial x} = - \frac{\partial L}{\partial x} - \lambda^T \frac{\partial f}{\partial x}; \quad \lambda^T(t_f) = \frac{\partial \phi}{\partial x}. \quad (A.7)$$

Equations (A.7) are called the co-state equations. Then, Eq. (A.6) becomes

$$\delta J = \lambda^T \delta x(t_0) + \int_{t_0}^{t_f} \frac{\partial H}{\partial u} \delta u dt. \quad (A.8)$$

For J to be minimum, δJ must be zero for arbitrary $u(t)$; this implies that for no bounds on u ,

$$\frac{\partial H}{\partial u} = 0, \quad t_0 \leq t \leq t_f \quad (A.9)$$

on the optimum path. If the control variables are constrained as

$$C(u, t) \leq 0, \quad (A.10)$$

then for $u(t)$ to be minimizing, we must have $\delta J \geq 0$ for all admissible $u(t)$. This implies, from Eq. (A.8) that

$$\frac{\partial H}{\partial u} \delta u \wedge \delta H \geq 0, \quad (A.11)$$

for all t and all admissible $\delta u(t)$. In other words, H must be minimized over the set of all possible u ; this is known as the minimum principle [7].

In summary, to solve for $u(t)$ that minimizes J , the differential equations (A.1) and (A.7) must be solved simultaneously, where $u(t)$ is determined from Eqs. (A.9) or (A.11). The boundary conditions on the state x at t_0 and λ at t_f are specified, resulting in a two-point boundary-value problem.

If L and f are not explicit functions of t , then

$$\begin{aligned}\dot{H} &= \frac{\partial H}{\partial t} + \frac{\partial H}{\partial x} \dot{x} + \frac{\partial H}{\partial u} \dot{u} + \dot{\lambda}^T f, \\ &= \frac{\partial H}{\partial t} + \frac{\partial H}{\partial u} \dot{u} + \left(\frac{\partial H}{\partial x} + \dot{\lambda}^T \right) f.\end{aligned}\tag{A.12}$$

Each element of Eq. (A.12) is zero on the optimum trajectory, from which we can conclude that H is constant on the optimum trajectory. This latter point is used in the analysis presented in Refs. 2 and 3.

Application to Aircraft Profile Optimization Using the Energy State Approximation

Here we are concerned with applying the above theory to the problem of choosing the thrust and airspeed values to control the aircraft vertical profile in going from one point to another. The cost function J is the direct operating cost (DOC) which is the sum of fuel and time costs. This is, in integral form,

$$J = \int_0^{t_f} (C_f \dot{w} + C_t) dt \stackrel{\Delta}{=} \int_0^{t_f} C_d dt,\tag{A.13}$$

where C_f is the cost of fuel (\$/lb), \dot{w} is fuel flow rate (lb/hr), C_t is the cost of time (\$/hr), C_d is the direct operating cost, and t_f is the time to fly the specified distance traveled d_f . It is also assumed that the typical vertical profile is as shown in Fig. A.1 - that is, it contains climb, cruise, and descent portions which have the constraint that

$$d_{up} + d_{dn} \leq d_f\tag{A.14}$$

where

$d_{up} = x(t_{cl})$ = the distance traveled from the start point to where the cruise segment begins (at time $t = t_{cl}$).

$d_{dn} = d_f - x(t_{cf})$ = the distance traveled from the end of cruise (at time $t = t_{cf}$) to where the descent segment ends.

ORIGINAL PAGE 13
OF POOR QUALITY

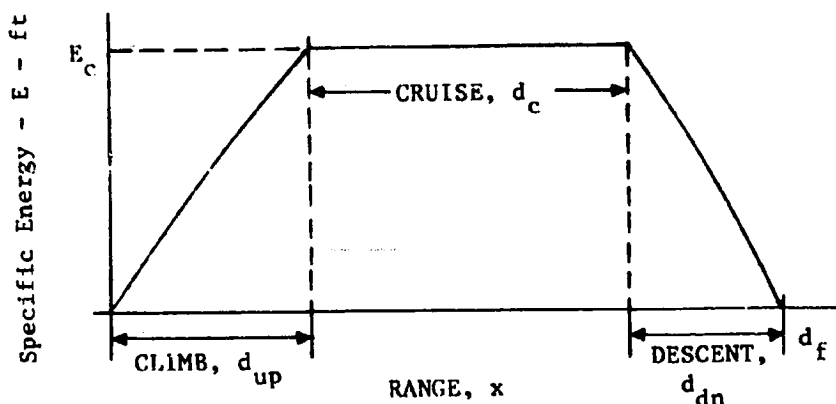


Figure A.1. Assumed Structure of Optimum Trajectories

Thus, the cost function (Eq. (A.13)) can be rewritten as

$$J = \int_0^{t_{c1}} C_d dt + (d_f - d_{up} - d_{dn})\psi + \int_{t_{cf}}^{t_f} C_d dt, \quad (A.15)$$

$$= \int_0^{t_{c1}} C_d dt + (d_f - x(t_{c1}) - [d_f - x(t_{cf})])\psi + \int_{t_{cf}}^{t_f} C_d dt.$$

where ψ is the cost per unit distance while in cruise.

Simplified point-mass equations of longitudinal motion of the aircraft are

$$\begin{aligned} \dot{V} &= (T-D)/m - g \sin \gamma, \\ \dot{h} &= V \sin \gamma, \\ \dot{x} &= V_g, \end{aligned} \quad (A.16)$$

where the flight path angle (γ) dynamics and weight loss due to fuel burn are neglected. Here,

- V = airspeed, (magnitude of aircraft velocity \bar{V} with respect to the air mass),
- V_g = ground speed (magnitude of $\bar{V}_g = \bar{V}_w + \bar{V} \cos \gamma$),
- \bar{V}_w = horizontal wind velocity,
- h = altitude,
- m = aircraft mass,
- T = thrust,

D = drag, and
x = horizontal range.

ORIGINAL PAGE IS
OF POOR QUALITY

Here, the range equation is based on the ground speed V_g (the vector sum of the horizontal velocity of the aircraft with respect to the air mass and the wind velocity). Also, it is assumed that lift $L = mg \cos \gamma$. The effect of weight loss is accounted for by continuously updating weight without adding another state variable.

The objective of this development is to simplify the optimization problem so that it can be solved in an on-board computer. This is done by use of the energy state approximation which is now presented [8]. Specific energy E is defined as

$$E = h + V^2/2g, \quad (A.17)$$

which is the sum of potential and kinetic energy per unit mass. Its time derivative is found to be

$$\dot{E} = V(T-D)/mg. \quad (A.18)$$

The energy state approximation is based on the assumption that potential and kinetic energy can be interchanged instantaneously. In this approximation, the energy state variable replaces altitude and airspeed state variables [8]. Thus, Eq. (A.17) can be used in place of \dot{V} and \dot{h} in Eq. (A.16).

It is assumed that the aircraft specific energy increases monotonically during climb and decreases monotonically during descent. This assumption is used in the development to change the independent variable in Eq. (A.15) from time to energy. This uses the transformation

$$dt = \frac{dE}{\dot{E}}. \quad (A.19)$$

It is mathematically convenient to evaluate the last integral in Eq. (A.15) backwards in time so that the energy state is monotonically increasing during its evaluation. This means that the running distance (range) variable during the descent can be measured backwards from the end point. Thus we can think of range measured in two ways as shown in Fig. A.2.

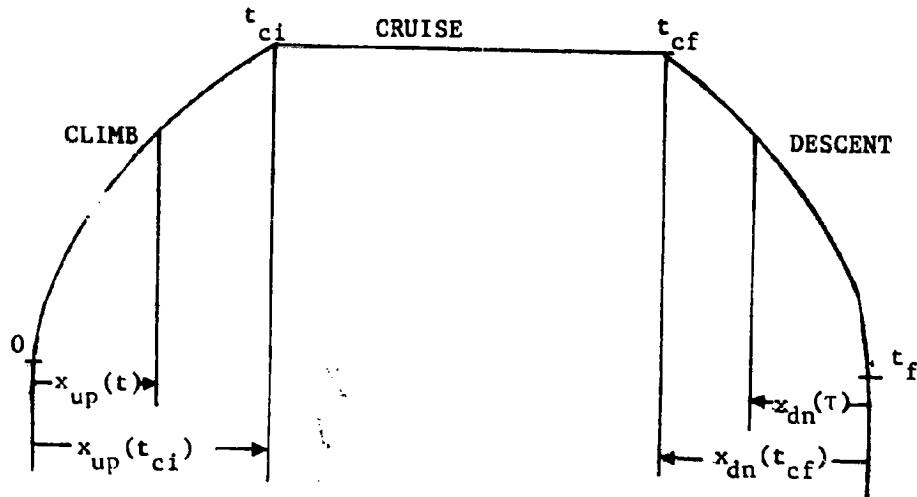


Figure A.2. Measurement of Range from the Origin or to the Destination.

In this sketch,

- $x_{up}(t)$ = range measured on the way up in forward time t ,
- $x_{up}(t_{ci})$ = value of x_{up} when initial cruise is reached,
- $x_{dn}(\tau)$ = range measured on the way up in backward time τ ,
- $x_{dn}(\tau_{cf})$ = value of x_{dn} when final cruise is reached ($\tau_{cf} = |t_{cf} - t_f|$).

Also, we define the variable x to be range traveled during climb and descent. The distance traveled during cruise is then constrained to be $(d_f - x)$. We can then see that an incremental change dx in the range variable x is equivalent to incremental changes in both x_{up} and x_{dn} . That is,

$$dx = d(x_{up} + x_{dn}) . \quad (A.20)$$

From this discussion, the second of Eqs. (A.15) can be written as

$$J = \int_{0_i}^{t_{ci}} C_d dt + (d_f - x_{up}(t_{ci}) - x_{dn}(\tau_{cf}))\psi + \int_{0_f}^{\tau_{cf}} |C_d| d\tau . \quad (A.21)$$

We use Eq. (A.19) and the transformation

$$dt = \frac{dE}{\dot{E}} \quad (A.22)$$

to rewrite Eq. (A.15) as

$$J = \int_{E_f}^{E_{ci}} \left(\frac{C_d}{\dot{E}} dE \right)_{\dot{E} > 0} + (d_f - (x_{up}(E_{ci}) + x_{dn}(E_{cf}))) \psi + \int_{E_f}^{E_{cf}} \left(\frac{dE}{|\dot{E}|} dE \right)_{\dot{E} < 0} \quad (A.23)$$

Here, E_i , E_{ci} , E_{cf} , and E_f are the values of specific energy evaluated at time t equal to 0 and t_{ci} and time τ evaluated at t_{cf} and t_f respectively.

Note from Eq. (A.23) that the range x only appears as the sum of climb and descent distances ($x_{up} + x_{dn}$). Thus, the state equation for this system of equations can be written as

$$\frac{dx}{dE} = \frac{d(x_{up} + x_{dn})}{dE} = \left(\frac{V_{gup}}{\dot{E}} \right)_{\dot{E} > 0} + \left(\frac{V_{gdn}}{|\dot{E}|} \right)_{\dot{E} < 0} \quad (A.24)$$

Here, V_{gup} and V_{gdn} are the equivalent ground speeds for climb and descent. Then, analogous to Eq. (A.4), the Hamiltonian is

$$H = \left[\left(\frac{C_d}{\dot{E}} \right)_{\dot{E} > 0} + \left(\frac{C_d}{|\dot{E}|} \right)_{\dot{E} < 0} + \lambda \left\{ \left(\frac{V_{gup}}{\dot{E}} \right)_{\dot{E} > 0} + \left(\frac{V_{gdn}}{|\dot{E}|} \right)_{\dot{E} < 0} \right\} \right] \quad (A.25)$$

This can be divided as

$$H = \left[\frac{C_d + \lambda (V_{gup})}{\dot{E}} \right]_{\dot{E} > 0} + \left[\frac{C_d + \lambda (V_{gdn})}{|\dot{E}|} \right]_{\dot{E} < 0} \quad (A.26)$$

Now, analogous to Eq. (A.7), the costate equation for λ can be written as

ORIGINAL PAGE IS
OF POOR QUALITY

$$\frac{\partial \lambda}{\partial E} - \frac{\partial H}{\partial x} - \frac{\partial H}{\partial (x_{up} + x_{dn})} = 0 \quad (A.27)$$

and from Eqs. (A.7) and (A.23), this costate has the final value

$$\lambda(E_{cl}) = \lambda(E_{cf}) = \frac{\partial \phi}{\partial (x_{up} + x_{dn})} = \frac{\partial ([d_f - x_{up} - x_{dn}] \psi)}{\partial (x_{up} + x_{dn})} = -\psi \quad (A.28)$$

where ψ is the cruise cost per unit distance.

Note, this problem could be placed in a slightly more conventional form by dividing it into two problems - one for climb and one-half of the cruise distance and the other for descent and the other half of the cruise distance. Then Eqs. (A.27) and (A.28) would be replaced by

$$\begin{aligned} \frac{\partial \lambda}{\partial E} - \frac{\partial H}{\partial x_{up}} &= 0 \quad (A.29) \\ \lambda(E_{cl}) &= \frac{\partial ([d_f/2 - x_{up}] \psi(E_{cl}))}{\partial x_{up}} = -\psi(E_{cl}), \end{aligned}$$

for climb. For descent,

$$\begin{aligned} \frac{\partial \lambda}{\partial E} - \frac{\partial H}{\partial x_{dn}} &= 0 \quad (A.30) \\ \lambda(E_{cf}) &= \frac{\partial ([d_f/2 - x_{dn}] \psi(E_{cf}))}{\partial x_{dn}} = -\psi(E_{cf}) \end{aligned}$$

This allows splitting the Hamiltonian defined in Eq. (A.26) and allows for $\lambda(E_{cl}) \neq \lambda(E_{cf})$. In fact, in the actual implementation $E_{cl} \neq E_{cf}$ because optimum cruise energy changes as fuel is burned off. The principal results are unchanged, however.

Thus, from Eq. (A.11), (A.29) and (A.30) the trajectory optimization problem becomes

ORIGINAL PAGE IS
OF POOR QUALITY

$$H_{up} = \min_{\substack{v_{up} \\ \pi_{up}}} \left[\frac{C_d}{\dot{E}} - \psi(E_{cf}) \left(\frac{v_{gup}}{\dot{E}} \right) \right]_{\dot{E} > 0} , \quad (A.31)$$

$$H_{dn} = \min_{\substack{v_{dn} \\ \pi_{dn}}} \left[\frac{C_d}{|\dot{E}|} - \psi(E_{cf}) \left(\frac{v_{gdn}}{|\dot{E}|} \right) \right]_{\dot{E} < 0} .$$

Thus, the optimization problem reduces to solving pointwise minimum values of the algebraic functions defined by Eq. (A.31) during the climb and descent portions of the trajectory.

Equations (A.29) and (A.30) are the transversality condition for the free final state problem ($d_{up} + d_{dn} < d_f$) with terminal cost. Thus, the constant value $c' \lambda$ for climb and descent is found to be the negative of the cost per unit distance for cruise.

The cruise cost $\psi (= -\lambda)$ is found by assuming that the aircraft is in static equilibrium during cruise ($T = D$), and that

$$\psi(E_c) = \min_{v_c} \left(\frac{C_d}{v_{gc}} \right) , \quad (A.32)$$

where

$$v_{gc} = |\bar{v}_c + \bar{v}_w| .$$

In other words, for any cruise altitude, there is an optimum thrust and airspeed V_c such that the cost per unit distance $\psi(E_c)$ is minimized. The optimum cruise cost as a function of cruise energy is typically of the shape shown in Fig. A.3. Thus, there is also an optimum cruise energy E_{copt} where cruise cost $\psi(E_{copt})$ is minimized. If the range is long enough so that there is sufficient range to reach optimum cruise energy E_{copt} during climb, it should be done, and the cruise conditions should be set so that $\lambda(E_c) = -\psi(E_{copt})$.

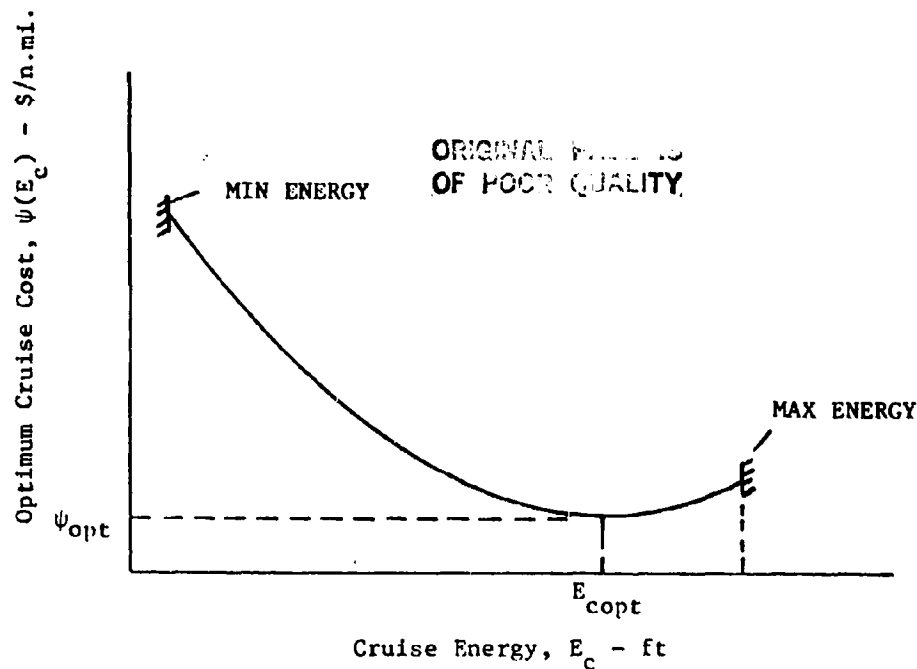


Figure A.3. Optimum Cruise Cost as a Function of Cruise Energy

For the case where there is no cruise segment ($d_f = d_{up} + d_{dn}$), the cost function contains only integral terms. Then, the transversality condition yields $\lambda = -\psi(t_c)$. That is, λ would be the negative of $\psi(t_c)$, where $\psi(t_c)$ is the optimum cost for cruising at the highest point reached on the climb trajectory.

The optimum cruise energy E_{copt} is only specifically reached when there is range enough to climb to and descend from the optimum altitude/airspeed values, where $\psi(E_c)$ is minimum. For ranges less than this value, the maximum value of E_c that is reached is a free variable less than the optimum value. Its choice is made to optimize the cost function of Eq. (A.23).

From Eqs. (A.23) and (A.25), one can write

$$\frac{\partial J}{\partial E} = H + \left[\frac{\partial[(d_f - d_{up} - d_{dn}) \psi(E_c)]}{\partial E} \right] = 0, \quad (A.33)$$

at $E = E_c$. This is

$$H_c + d_c \frac{\partial \psi}{\partial E_c} = 0 \quad ,$$

ORIGINAL PAGE IS
OF POOR QUALITY

(A.34)

where d_c is the cruise distance, and H_c is the total value of H ($H_{up} + H_{dn}$) at the cruise point. Thus, Eq. (A.34) can be used, along with other characteristics of ψ and H , to determine the relationship between ψ , E_c , and d_c . The Hamiltonian evaluated at $E = E_c$ is the cost penalty to achieve a unit increase in cruise energy. For $H_c > 0$, Eq. (A.34) can be written as

$$d_c = -H_c / (\partial \psi / \partial E)_E = E_c \quad (A.35)$$

Figure A.4 shows the family of trajectories which have this characteristic. These occur at values of E_c below $E_{c_{opt}}$ where $\partial \psi / \partial E < 0$ (see Fig. A.3). That is, non-zero cruise segments occur at short ranges with cruise energies less than the optimum energy value for long range.

For the case where $H_c = 0$, d_c is zero for $\partial \psi / \partial E < 0$. The distance d_c can be non-zero only at optimum cruise energy where $\partial \psi / \partial E = 0$. This family of trajectories is shown in Fig. A.5.

Thus, we have a situation where positive values of H_c dictate one type of trajectory and zero values dictate another. In Ref. 2, it is shown that if the aircraft engine specific fuel consumption S_{FC} is independent of the thrust T (so that $\dot{w} = S_{FC}T$), then the structure of the trajectories will be like Fig. A.5 with no cruise segment occurring except at $E_{c_{opt}}$. (This implies that the Hamiltonian H_c is zero at the maximum energy point). For this case, the optimum thrust setting for climb is T_{max} , and the optimum setting for descent is T_{idle} .

If the engine specific fuel consumption is dependent on thrust, and the thrust values are not constrained during climb or descent, it is shown in Ref. 2, that the Hamiltonian H_c is again zero at the cruise energy, and again the trajectory structure is like those of Fig. A.5. S_{FC} is dependent on thrust for the aircraft models used in the OPTIM program.

ORIGINAL PAGE IS
OF POOR QUALITY

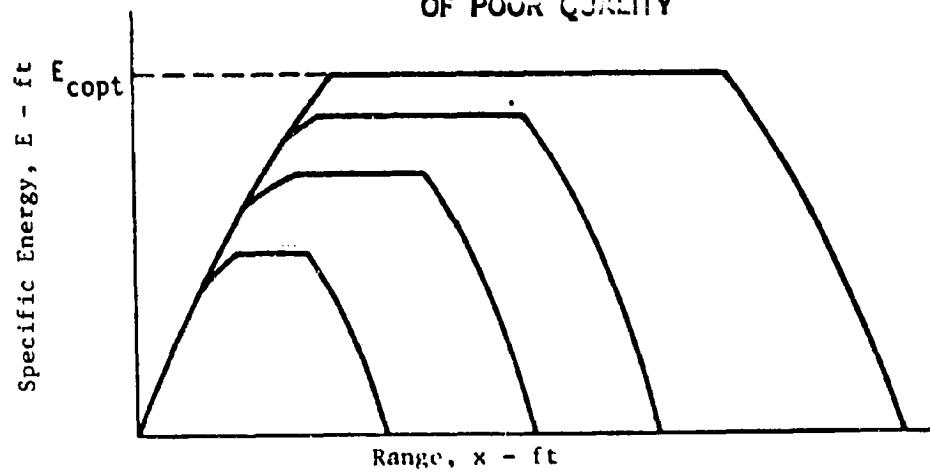


Figure A.4. Optimum Profile Energy vs Range for $H_c > 0$ at Cruise.

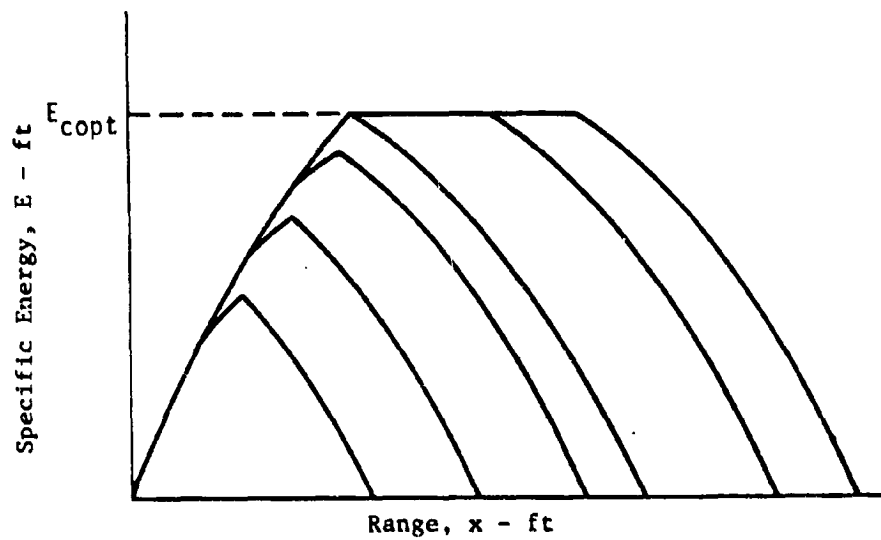


Figure A.5. Optimum Profile Energy vs Range for $H_c = 0$ at Cruise.

If the S_{FC} is dependent on thrust, and constrained to the maximum value for climb and to the minimum idle value for descent, then the Hamiltonian is positive at cruise. This causes positive cruise segments according to Eq. (A.35) at cruise energies below the optimum. For this case, the optimum trajectories will have shapes similar to Fig. A.4. These trajectories are slightly less efficient than those of Fig. A.5. because one less control is available for optimization.

Some Mechanization Details of the Computer Program

The remaining sections of this Appendix describe how the previous theoretical material has been utilized to construct an offline computer program for generating optimum vertical profiles for models of two turbo-jet aircraft provided by NASA. This material is presented in an alternate way in Ref. 3, and the program is referred to here as OPTIM.

By examining the specific fuel consumption data of the turbojet engine models, it was determined that S_{FC} is dependent on thrust. Thus, two types of short range profiles must be considered - those represented by Fig. A.4 (Type 1 profile) when thrust is constrained and airspeed is the single control - and those represented by Fig. A.5 (Type 2 profile) when both thrust and airspeed are used as controls.

The solution to optimum climb and descent profiles is found by minimizing the Hamiltonian expressed in Eqs. (A.31). The independent variable (energy) is stepped along in fixed increments (e.g., 500 ft), and the Hamiltonian is minimized at each energy setting. Minimization occurs by finding the best values of airspeed (V_{up} , V_{dn}) and possibly thrust (π_{up} , π_{dn}) so that the climb function and the descent function are individually minimized.

To solve Eqs. (A.31) requires knowing two more quantities:

$\psi(E_c)$ - the cruise cost per unit distance. This comes from evaluating Eq. (A.32) at the desired cruise altitude.

ORIGINAL PAGE IS
OF POOR QUALITY

E_c - the cruise energy. This is a function of the cruise altitude and the associated cruise airspeed obtained in Eq. (A.32).

Note that for the Type 2 profile at short ranges, there is no cruise segment. In this case, the maximum energy achieved at maximum altitude is referred to as the cruise energy E_c . At that altitude, there still is defined a minimum cruise cost according to Eq. (A.32).

For the Type 1 trajectory of short range, there exists a non-zero cruise segment which is determined by use of Eq. (A.35). To solve Eq. (A.35) requires that the Hamiltonian defined by Eqs. (A.31) be solved at the point of transition from climb-to-cruise. It also requires knowing the slope $\partial\psi/\partial E$ of the cruise cost for a change in cruise energy at that point.

Cruise Optimization

The first step that must be taken to compute optimum trajectories is to derive the optimum cruise cost ψ and its derivative $\partial\psi/\partial E$. This is done by computing what is referred to as the "cruise table". The parameters that affect this table are the assumed cruise weight, the wind profile, and the lift L , drag D , thrust T , and fuel flow rate \dot{w} characteristics of the aircraft. The optimization process searches over the acceptable ranges of altitude and airspeed for a given weight. The results are collected in tabular form for a series of different assumed cruise weights.

Again, the minimum cost of flight during cruise per unit distance for a fixed cruise weight W_c is found by

$$\psi(W_c) = \min_v \left[\frac{C_f \dot{w} + C_t}{|\bar{v}_{gc}|} \right] \quad (A.36)$$

This assumes that the aircraft is in static equilibrium during cruise, i.e.,

$$T \cos \alpha = D, \quad (A.37)$$

$$L + T \sin \alpha = W,$$

where the angle-of-attack α is found by solving these equations simultaneously. The altitude is stepped in 1000 ft increments from 10,000 ft to ceiling altitude (where maximum thrust just balances drag). At altitudes below ceiling altitude, the airspeed - dependent drag curve crosses the maximum thrust curve at two points (V_1 and V_2) as illustrated in Fig. A.6. Thus, for each altitude level, the values of V_1 and V_2 are determined, and then $\psi(W_c, E_c)$ is minimized with respect to airspeed V between these two limits. Restrictions are that V_1 be greater than 0.1 Mach and that V_2 be less than 0.89 Mach or 0.84 Mach for buffet constraint reasons.

After the cruise cost is minimized at each discrete altitude level, these numbers are stored in a table with altitude as the independent variable. Typical results are plotted in Fig. A.7. Presented here are also the optimum cruise Mach number M_{opt} and the optimum thrust setting EPR_{opt} . After results are obtained in steps of 1000 ft, the minimum cost point is found as a function of altitude. In the OPTIM program, the cruise table optimization results are obtained by using a Fibonacci search with eight Fibonacci numbers.

The cruise table results are obtained for cruise weights varying as dictated by the program input. Usually, the cruise weight is incremented in steps of 5000 lb. Up to ten values of cruise weight can be used. For each cruise weight, the optimal cruise altitude, cost, speed, power setting, fuel flow rate, and specific energy are computed. An example of optimum cruise cost as a function of cruise weight is shown in Fig. A.8.

Climb Optimization

After the cruise tables are generated, the program proceeds with obtaining the optimum climb trajectory. This requires guessing what the cruise weight will be, based on the takeoff weight. This guess is used to obtain a trial value for ψ_c (or λ) in the Hamiltonian from

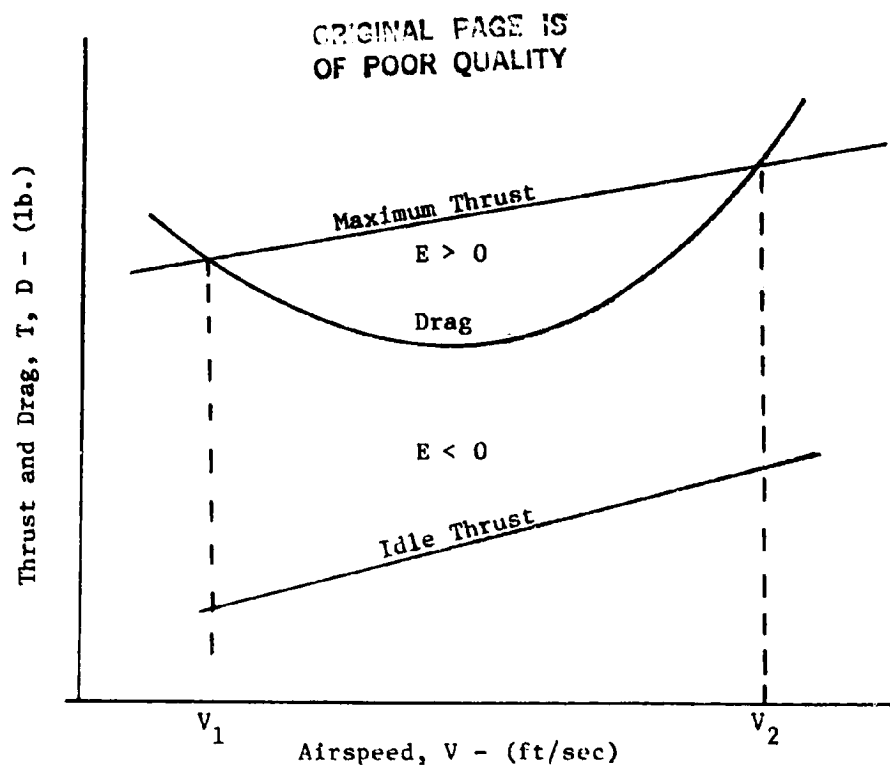


Figure A.6. Plot of Thrust and Drag vs Airspeed at a Particular Altitude

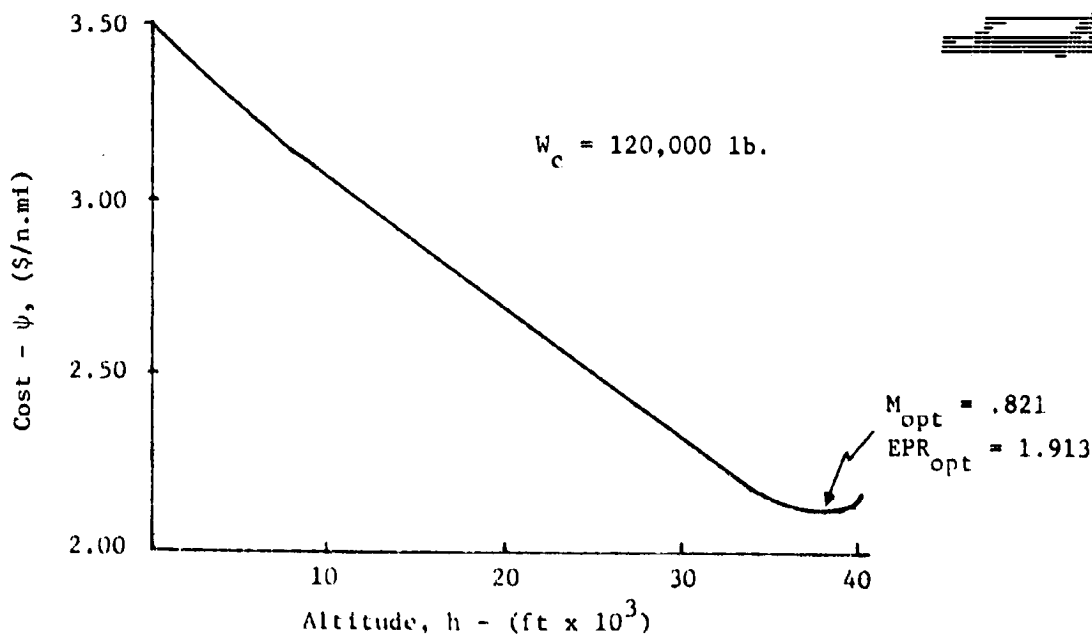


Figure A.7. Optimum Cruise Cost as a Function of Altitude for Cruise Weight of 120,000 lb.

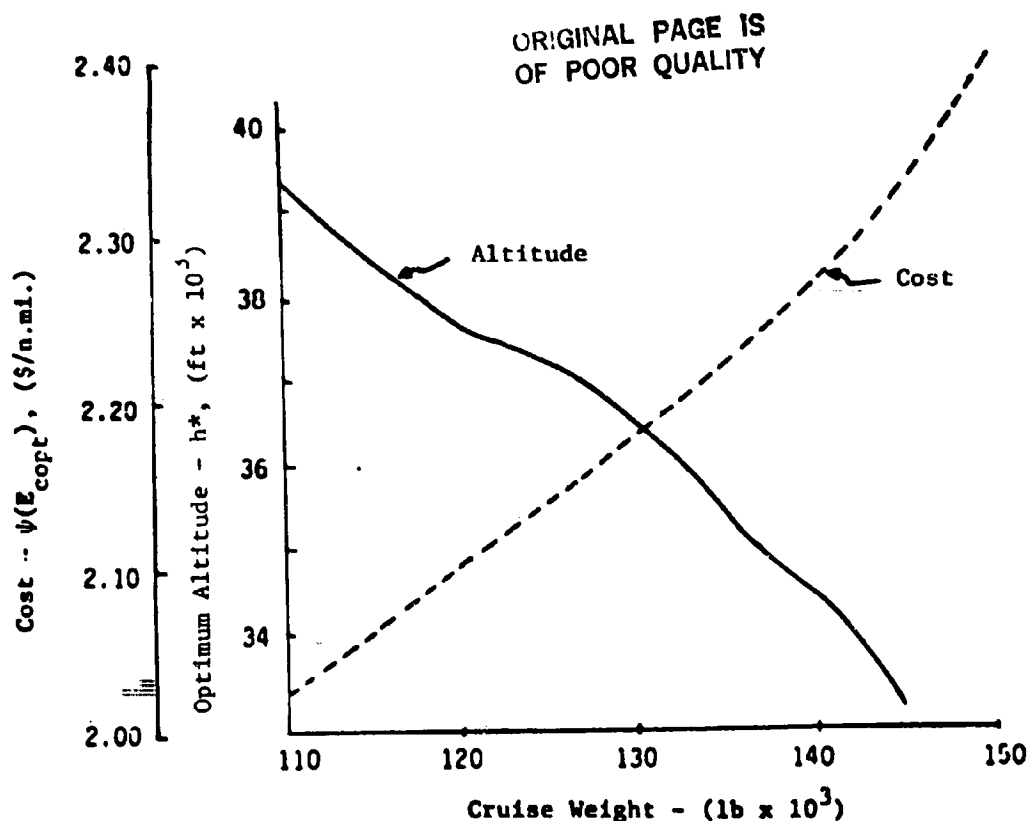


Figure A.8. Optimum Cruise Cost and Cruise Altitude as Functions of Cruise Weight for the Tri-jet Aircraft Flying into a Particular Head Wind.

the cruise tables. The procedure to obtain this guess is based on an empirical formula which iterates until convergence is made.

The climb optimization process starts by assuming $\lambda(E_c) = 1.01 \psi_c(E_{c_{opt}})$ or $1.0 \psi_c(E_{c_{opt}})$, where $\psi_c(E_{c_{opt}})$ is first obtained by setting the initial cruise weight W_{c1} equal to the takeoff weight (an input). The appropriate cruise tables are used to interpolate to find the corresponding value of E_c associated with $1.01 \psi_c$ or $1.0 \psi_c$. Then, empirical equations are used to obtain an approximation to the fuel burned to reach E_c . For example, for the tri-jet model, the form is

$$F_{up} = 0.11 (E_{c1} - E_i) (1 + 4.7 C_t/C_f) W_{c1}/W_{ref}. \quad (A.38)$$

Here, E_i is the takeoff aircraft energy, W_{ref} is a reference weight (136000 lb) for the tri-jet, and W_{c1} is the previous value of cruise weight. Then,

ORIGINAL PAGE IS
OF POOR QUALITY

the cruise weight is updated at $W_{ci} = W_{ci} - F_{up}$. This process is repeated until the difference in consecutive estimates of F_{up} falls below 100 lb.

When the cruise weight estimate is obtained, the corresponding values of E_c and $\lambda(E_c)$ are obtained from the cruise tables. Then, the program is ready to generate points on the optimum climb trajectory. This is done by stepping along at discrete increments of specific energy and minimizing the Hamiltonian function

$$H_{up}(E) = \frac{C_f \dot{W} + C_t - \lambda(E_c) (V + V_w)}{\dot{E}} \quad (A.39)$$

at each point. (This is the first of Eqs. (A.31)). That is, the program starts with initial energy $E_o = h_o + V_o^2/2g$. It steps the energy a fixed amount ΔE (say 500 ft). At this point, it searches over airspeed V (and possibly thrust setting π) so that Eq. (A.39) is minimized. For the turbojet engines, thrust is governed by EPR settings which vary between 1.1 (idle thrust) and some maximum value less than 2.4. The airspeed has an upper limit governed by

- a). 0.89 or 0.84 Mach (buffet limits),
- b). 250 kt (IAS) below 10,000 ft for ATC restrictions,
- c). $\sqrt{2g(E-h)}$ which insures that the aircraft climbs, and
- d). V_2 , the upper value shown in Fig. A.6 where max thrust equals drag.

The lower limit is governed by

- a). V_1 , the lower value shown in Fig. A.6 where max thrust equals drag,
- b). 0.1 Mach
- c). 5 ft/sec less than the previous value of V to limit large jumps in flight path angle.

The Fibonacci search technique is again used to determine V and π which minimize Eq. (A.39) for the fixed value of energy E . The value chosen for airspeed is accurate to within .0056 Mach, and EPR is accurate

to within .009. Associated with these values of V and π are values of energy rate \dot{E} (Eq. (A.16)) and altitude h :

$$h = E - V^2/2g. \quad \text{ORIGINAL PAGE IS OF POOR QUALITY} \quad (A.40)$$

From these, approximate values of time, range, flight path angle, and fuel burned are obtained from

$$\begin{aligned} \Delta t &= \Delta E / \dot{E}, \\ \sin \gamma &= (\Delta h / \Delta t) / V, \\ x &= \sum \Delta x; \quad \Delta x = V_g \Delta t \\ F &= \sum \Delta F; \quad \Delta F = w \Delta t. \end{aligned} \quad (A.41)$$

The ground speed V_g is found as

$$V_g = V_w \cos(\psi_g - \psi_w) + \sqrt{V_w^2 \cos^2(\psi_g - \psi_w) + V^2 - V_w^2}, \quad (A.42)$$

where ψ_g and ψ_w are the desired aircraft ground heading and wind heading, respectively.

The above process is repeated by stepping along energy in increments of ΔE until E_c is reached. The last value of Eq. (A.39) is stored for possible use in evaluating the cruise distance.

≡ The above climb optimization procedure is repeated with λ set to various multiples of the optimum cruise value ψ_c until the total range of flight converges to the appropriate value. This is discussed in further detail later.

Descent Optimization

The descent optimization is very similar to the climb optimization with regard to the equations which are evaluated. The optimization process requires estimated values of E_c and W_c at the beginning of descent, and an estimate of weight W_f at the end of descent. The method used to obtain these estimates is discussed in the next section.

If there is a cruise portion of flight, fuel will be burned off during cruise. Thus, the value of E_{cf} , ψ_c , and W_{cf} at the beginning of descent will be different than at the beginning of cruise. If there is no cruise portion, then these values will be identical.

ORIGINAL PAGE IS
OF POOR QUALITY

The descent profile is obtained by starting at the final energy state and then going backwards in time. The energy rate is constrained to be negative with respect to forward time.

Similar descent profile constraints exist on airspeed as for those of the climb profile. The thrust level is on or near the idle value during descent.

Cruise Fuel Burn

To estimate the final weight during cruise (W_{cf}) and landing (W_f), the following steps are taken:

- 1). Determine ψ_c , the initial cruise cost based on the initial cruise weight W_{ci} obtained from the climb optimization.
- 2). Use the initial cruise weight to compute the fuel flow rate $\dot{w}(\psi_c)$ from the cruise table.
- 3). Estimate the cruise range d_c by use of empirical equations; i.e., for the tri-jet model:

$$P = \psi_c / \psi_{copt} = 1.0 \text{ or } 1.01, \quad (A.43)$$

$$d_c = b_1 P^4 + b_2 P^3 + b_3 P^2 + b_4 P + b_5.$$
- 4). Compute the cruise fuel as

$$F_c = \dot{w}(\psi_c) d_c / V_{gc}. \quad (A.44)$$
- 5). Estimate the average cruise weight as

$$\bar{W}_c = W_c - 0.5 F_c \quad (A.45)$$
- 6). Use the cruise table to obtain the corresponding cruise cost $\bar{\psi}_c$, altitude \bar{h}_c , fuel flow rate $\dot{w}(\bar{\psi}_c)$, airspeed \bar{V}_c , and wind speed $\bar{V}_w(\bar{h})$.
- 7). Recompute Eq. (A.44) and then find the final cruise weight,

$$W_{cf} = W_{ci} - F_c. \quad (A.46)$$
- 8). Use the value W_{cf} in the cruise tables to obtain $\psi(W_{cf})$. As with the climb, set $\lambda = 1.0 \psi_c(W_{cf})$ or $1.01 \psi_c(W_{cf})$.

- 9). Use this value of λ to obtain h_{cf} and E_{cf} from the cruise tables. These are the end conditions for the descent trajectory obtained backwards in time.
- 10). Estimate the landing weight from empirical formulas; e.g., for the tri-jet:
- $$P = 1.01 \text{ or } 1.0 \quad (A.47)$$
- $$W_f = W_{cf} - (c_1 P^2 + c_2 P + c_3).$$

The values of λ , E_{cf} and W_f obtained by the above procedure are used for obtaining the optimum descent trajectory. The descent portion of the Hamiltonian is of the form

$$H_{dn}(E) = \frac{C_f \dot{w} + C_t - \lambda (E_{cf}) (V_g)}{|E|}; \quad (A.48)$$

this function is also minimized at each of the given values of energy.

After the first descent profile is completed, a new estimate of cruise distance is obtained by using Eq. (A.35), or

$$d_c = -(H_{up} + H_{dn}) / (\partial \psi / \partial E). \quad (A.49)$$

Then, step (4) above is repeated to obtain an improved cruise fuel burn. Then, the improved landing weight estimate is

$$W_f = W_i - (F_{up} + F_c + F_{dn}). \quad (A.50)$$

The landing trajectory is reoptimized with this new value of landing weight. Then, improved values of total range traveled, time required, and fuel burned during climb, cruise, and descent are made.

For short range flight, the above steps assumed that a Type 1 trajectory is generated because thrust is constrained to maximum value during climb and idle value during descent. If thrust is free, then a Type 2 ~~trajectory will result~~ trajectory will result, with no cruise portion. For this case, the steps required to estimate cruise distance d_c and final cruise cost, weight, and energy can be eliminated.

Cruise Cost Estimation

The first climb and descent profiles are generated with $\lambda = \psi_c(E_{\text{copt}})$ for free thrust or $\lambda = 1.01 \psi_c(E_{\text{copt}})$ for fixed thrust. The resulting range of the optimum profile has a given value defined as R_{max} . An example is shown for $\lambda = 1.01 \psi_c(E_{\text{copt}})$ in Fig. A.9. If the total range desired is greater than R_{max} , then the climb associated with $\lambda = \psi_c(E_{\text{copt}})$ is used, the required cruise distance is computed, and the descent profile is recomputed to produce the correct overall range.

If the desired range is less than R_{max} , λ is next set to a value corresponding to a cruise altitude just over 10,000 ft. (This typically has the value $\lambda = 1.3 - 1.5 \psi_c$). The optimum profile is recomputed, and the associated range is referred to as R_{min} . (See Fig. A.9)

If the desired range is between R_{min} and R_{max} , then an iterative process is used to obtain $\psi(E_c)$ and the associated desired range. Iterations are stopped when the total range traveled is within some ϵ of the desired range. (In OPTIM, ϵ is set at 5 n.mi.)

If the cruise altitude is fixed, or a step climb is used between two fixed altitudes, then λ is set to $\psi(E_{\text{copt}})$ corresponding to the fixed altitude. In this case, no iteration on λ is required.

ORIGINAL PAGE IS
OF POOR QUALITY

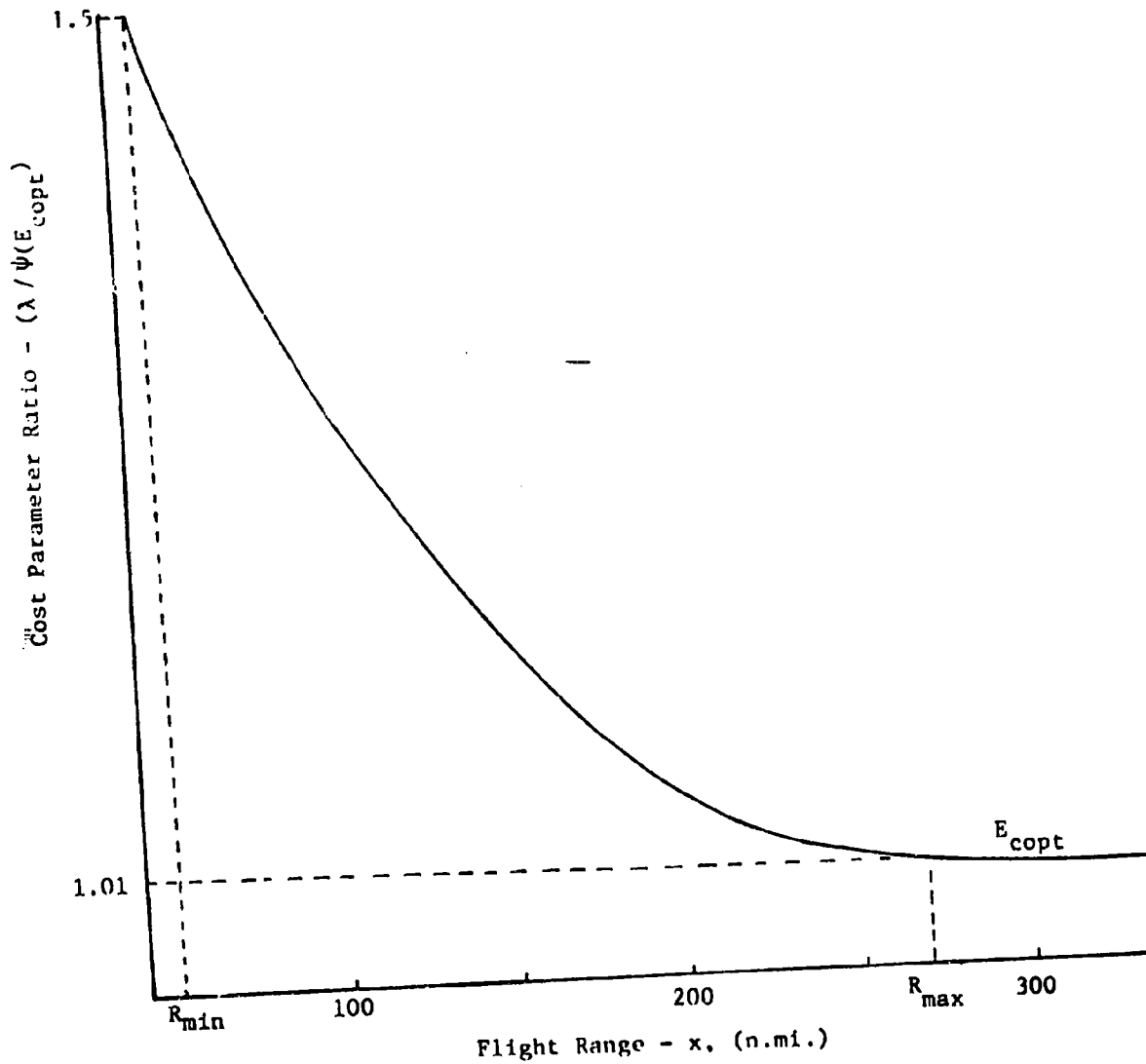


Figure A.9. Relationship between the Cruise Cost Parameter ψ and the Associated Range of Flight

APPENDIX B

OPTIM SUBROUTINE DESCRIPTION

This section contains an explanation of the data storage for program OPTIM. Following that is an explanation of the main program, its two principal subprograms, and then the remaining subroutines and functions in alphabetical order.

Data Storage

The major part of the data communications between subroutines in OPTIM is through labelled common statements. There are ten such commons. Their names and a short description of each are:

CCDE	Cruise, climb, descent variables.
CONST	Constants.
CRUISE	Cruise table and associated variables.
DESCRP	Assorted variables.
ERROR	Error traceback information.
GRAPH	Data to be written to Unit 11 and associated variables. Includes the final climb and descent trajectories.
INPUT	All input parameters.
TOA	Time-of-arrival and step climb variables.
TRIJET	Engine data, tri-jet.
TWINJT	Engine data, twin-jet.
WINDP	Wind input data and associated parameters.

As a convenience, the CDC UPDATE capability is used to insert COMMON statements into source decks. This facilitates changing items in COMMON with no loss of program portability, because UPDATE produces a compile file which is directly listable, editable, and compilable by any standard FORTRAN.

Subroutines

Following are more detailed descriptions of the main program OPTIM and its subroutines and functions.

MAIN PROGRAM: OPTIM

This program synthesizes a fixed range, minimum fuel or direct operating cost trajectory. The overall process is explained in Section IV and Appendix A. The main program calls a subroutine (ALLIN) which reads all required input, and then calls the principal control subroutine (OPTM56).

Subroutines called:

ALLIN
OPTM56

Commons used:

None

OPTM56

OPTM56 controls the main program computations. It first calls CHEKIN to check that all input quantities are within reasonable limits. It may then call OPTTOA to execute a run with fixed time-of-arrival, if ICTAB = 2. Otherwise, it calls in sequence PKOFIL to compute the optimum requested profile, STEP to adjust the trajectory to include a step climb, if appropriate, and PRFTBL to output the results of the optimization. If any error has occurred during the computation, TRACIT is called to output the traceback information.

Subroutines called:

CHEKIN
OPTTOA
PRFTBL
PROFIL
STEP
TRACIT

Commons used:

CCDE
CRUISE
DESCRP
ERROR
GRAPH
INPUT

OPTTOA

This subroutine serves as the program executive when synthesizing a vertical flight path which requires minimum fuel to achieve a fixed range in a fixed time-of-arrival. The logic is similar to that of OPTM56 except the following items are changed:

1. An outer loop structure is mechanized to iterate on TC to achieve the desired time-of-arrival TEND. This is illustrated by the block diagram in Fig. 4, Chapter IV. TC is initially set to zero, and the minimum fuel trajectory is generated. The time this trajectory takes, $TTIME_0$, is compared to TEND, and the subsequent logic depends on whether $TTIME_0$ is greater or less than TEND.
2. If $TTIME_0$ is less than TEND, a special trajectory is next generated with the cruise speed set at that value where minimum fuel rate ($\max/L/D$) is achieved. This represents the upper bound on length of flight time that is practical without using path stretching or a holding pattern. If this trajectory takes $TTIME_1$, then TEND is compared to $TTIME_1$. If TEND is greater than $TTIME_1$, the program stops. If $TTIME_0 < TEND < TTIME_1$, the cost of time TC is negative, and iterations on TC continue until $TTIME_n$ is within 10 sec of TEND.
3. If $TTIME_0$ is greater than TEND, the cost-of-time TC is positive. Iterations on TC continue until $TTIME_n$ is within 10 sec of TEND.

The logic uses the previous three values of TC to fit a quadratic curve to $TTIME$ as a function of TC. The desired value TEND is then used to obtain the next trial value of TC. The program stops if convergence has not been achieved after nine trials.

Subroutines called:

MATINV
PRFTBL
PROFIL
STEP

Commons used:

CCDE
CONST
CRUISE
DESCRP
ERROR
INPUT
TOA
WINDP

ORIGINAL PAGE IS
OF POOR QUALITY

ALLIN

ALLIN reads all input, including the cruise table if necessary. It calls WINDIN to read the wind in. ALLIN also outputs the input data and lists the options chosen.

Subroutines called:

BANNER
PAGE
SERCH1
TRACIT
WINDIN

Commons used:

CCDE
CRUISE
CONST
DESCRP
ERROR
INPUT
WINDP

ATLOW

This subroutine generates the atmospheric density (in $\text{lb sec}^2/\text{ft}^4$), atmospheric pressure (in lb/ft^2), atmospheric temperature (in degrees Kelvin) and speed of sound (in ft/sec) at a given altitude below 20,000 meters (65,617 feet). It also makes the appropriate modifications in atmospheric density and speed-of-sound to account for variations in standard day temperature (represented by the input DTEMP). The 1962 standard atmosphere is used. This version of the program does not calculate a new atmosphere when called at successive times at the same altitude.

BANNER

ORIGINAL PAGE IS
OF POOR QUALITY

BANNER is a subroutine used to write the title, date, and time of day at the beginning of a run.

Subroutines called:

DATE }
TIME } CDC-supplied

Common used:

None

BLOCK DATA - DATTRI

This block data contains the engine data used with the tri-jet aircraft model. Three tables are used to describe idle thrust, idle fuel flow, and maximum continuous engine pressure ratio (EPR).

Subroutines called:

None

Common used:

TRIJET

BLOCK DATA - DATTWN

This block data contains numerical characteristics of the turbofan engine used with the twin-jet aircraft model. Seven tables are used to describe idle thrust and fuel flow for bleed valves open and closed, altitude of surge bleed valve closure, maximum EPR for climb and cruise, and Mach number corrections.

Subroutines called:

None

Common used:

TWINJT —

CDRAG

This subroutine calls the appropriate routine to compute the aircraft drag coefficient CD based on the particular aircraft model selected by the input variable IAC. Currently, two models are available, but logic is present to use up to four different aircraft.

Subroutines called:

CDRAG1*
CDRAG2
CDRAG3
CDRAG4*

Commons used:

None

* not included with program

CDRAG2

This subroutine computes the drag coefficient CD for some given Mach number EM and lift coefficient CL for a medium range tri-jet transport aircraft model. The value is computed from the coefficients of a polynomial stored in the array COEFF.

Subroutines called:

POLY2

Commons used:

None

CDRAG3

This subroutine computes the drag coefficient CD for some given Mach number EM and lift coefficient CL for a medium range twin jet transport aircraft model. CD is computed by polynomial evaluation, including interpolation of the polynomial and its first derivative in certain regions, as necessary.

Subroutines called:

POLY2

Commons used:

None

CHEKIN

CHEKIN tests several of the input quantities, such as the weight and time inputs, to ensure that they are within reasonable limits.

Subroutines called:

None

Commons used:

INPUT

CLIFTT

CLIFTT calls the appropriate routine to compute the aircraft lift coefficient for the particular aircraft model selected by the input variable IAC. Currently, two models are available, but logic is present to use up to four different aircraft.

Subroutines called:

CLIFT1*

CLIFT2

CLIFT3

CLIFT4*

~~XXXXXXXXXX~~ Commons used:

None

* not included with program.

CLIFT2

This subroutine computes the lift coefficient CL for a medium range tri-jet transport aircraft as a function of Mach number EM, altitude H, and angle-of-attack ALPHAP. The lift coefficient consists of three terms:

$$C_L = C_L \text{ (basic)} + C_{L0} + C_{L\alpha} \alpha$$

The first term C_L (basic) is a polynomial function of angle-of-attack α . The value of this term is checked against the buffet boundary expressed as a polynomial of Mach number. The second term C_{L0} is a polynomial of Mach number with altitude as the parameter. The third term $C_{L\alpha}$ is also a polynomial of Mach number. The coefficients of the polynomial are fit for different altitudes.

Subroutines called:

POLYB

Commons used:

None

CLIFT3

This subroutine computes the lift coefficient CL for a medium range twin jet transport aircraft as a function of Mach number EM, altitude H, and angle-of-attack ALPHA. The lift coefficient consists of three terms:

$$C_L = C_L \text{ (basic)} + C_{L0} + C_{L\alpha} \alpha$$

The first term C_L (basic) is a function of angle-of-attack. The second term C_{L0} is a function of altitude and Mach number. The third term $C_{L\alpha}$ is also a function of altitude and Mach number. These terms are determined by table lookup.

Subroutines called:

DBLSRC
SERCHI

Commons used:

None

CLIMB

This subroutine controls the computation of the climb portion of the flight profile. CLIMB calls FULEST to estimate the fuel use during climb. It then uses the associated λ from the cruise table computation to optimize the climb trajectory. The optimization is calculated in FCLMB6 at a series of energy steps. The energy step size is computed in ESTEP, and the incremental time, distance, fuel used, and associated quantities are computed in STEPEN at each step. All climb variables are stored at each step for later output and, if desired, plotting.

After the climb is completed, the actual fuel use is found. If the estimated and actual fuel use differ by more than 200 pounds, the climb is recomputed with the λ associated with the new weight. Only one iteration of climb is permitted.

Subroutines called:

ATLOW
ESTEP
FCLMB6
FIAS
FULEST
ICLOCK
PRWT
SGLSRC
STEPEN
WLEFHV

Commons used:

CCDE
CONST
DESCRP
ERROR
GRAPH
INPUT
WINDP

CRUISR

CRUISR is used in the step climb option only. It is assumed that the aircraft cruises from weight W_1 to W_2 (with W_1 being greater than W_2 , in the initial pre-step climb cruise table). CRUISR returns cost, energy, fuel use rate and Mach number at the new cruise weight W_2 . It also computes distance travelled in going from W_1 to W_2 .

Subroutines called:

SERCHD

Commons used:

CCDE
CRUISE
DESCRP
ERROR
TOA

CRUISX

CRUISX looks up in the cruise table, and returns the flight parameters associated with crusing for a horizontal distance of RANGEX, starting at weight W_1 . On later entries, it computes the effect of crusing an increment of RANGEX from the previously computed position (without looking up a new weight).

Subroutines called:

CTABLE
SERCHD
SERCHI

Commons used:

CCDE
CRUISE
DESCRP
ERROR

This subroutine generates the cruise table. The cruise table includes the minimum cruise cost $\psi(W)$, altitude h , and airspeed V as functions of weight W . For each given weight W , the optimum cruise cost, altitude and speed are computed as follows:

1. For some given altitude h :

- a) determine the minimum drag speed V_{mD} ,
- b) compute the maximum thrust T_{max} available for this speed and altitude,
- c) if T_{max} is greater than D and V_{mD} proceed; otherwise the ceiling has already been reached, and proceed to 2.
- d) compute the lower and upper permissible speeds V_a and V_b , where the maximum thrust curve intersects the drag curve.
- e) minimize the following cost function,

$$\psi = \min_{V_a, V_b} \frac{C_f \dot{f} + C_t}{V_g},$$

where

- C_f = the fuel cost in \$/lb,
 \dot{f} = fuel flow rate in lb/hr,
 C_t = time cost in \$/hr,
 V_g = ground speed in ft/sec.

Steps (a)-(c) are repeated for all permissible altitudes ranging from 10,000 ft to either HMAX or the maximum ceiling. For a fixed altitude run, values are calculated for only the input altitude. For fixed TOA, the minimum altitude is 20,000 ft.

2. Determine the minimal cruise cost $\lambda^*(W)$ and its associated h and V as follows:

$$\lambda^*(W) = \min_h \psi$$

(For a fixed altitude run, this minimization step is omitted.)

Steps (1) and (2) are repeated for different weight values starting with WEIGHT and decreasing in steps of DEW through WN.

During a fixed time-of-arrival run, on the second iteration (IMFD=1), this subroutine performs the same process to determine the airspeed at each altitude and cruise weight where fuel flow rate \dot{f} is minimized. This occurs at near maximum lift/drag.

The output of CRZOP5 may be written on Unit 8 which can then be stored for later use. The output is also printed on Unit 6.

Subroutines called:

FIASM
FMIN2
FOPT
JTRUNC
LSQPOL
PAGE
SERCHI

Commons used:

CCDE
CRUISE
CONST
DESCRP
ERROR
INPUT
WINDP
TOA

CTABLE

CTABLE looks up and returns the set of parameters in the cruise table corresponding to weight WCRUZ.

Subroutines called:

SERCHD

Commons used:

CCDE
CRUISE
DESCRP
ERROR

ORIGINAL PAGE IS
OF POOR QUALITY

DBLSRC

This function performs a double table lookup. Given a function $f(x,y)$, this function interpolates the appropriate arrays to obtain approximate values of $f(A,B)$. The four points which surround (A,B) are first found, and the function is evaluated at each. Then these values are interpolated, first on x and then on y , to obtain the approximate solution.

Subroutines called:

SERCHI

Commons used:

None

DESCND

DESCND controls the computation of the descent portion of the flight profile. It first calls WATEST to estimate the remaining cruise range, cruise fuel, and descent fuel. After this point, the computation is similar to that of CLIMB. ESTEP specifies energy step increments, at each of which the trajectory is optimized within FDESCN6. STEPEN is called to calculate additional dependent parameters at each step, and all quantities are stored for later output and retrieval. In addition, DESCND computes cumulative descent cost and total cost (climb plus descent) at each step.

If DESCND has been called during a constrained descent (DESPC), it saves in COMMON the value of the altitude and fuel use at which the unconstrained descent crosses the critical altitude, HCABSL.

Subroutines called:

CTABLE
ESTEP
FDSCN6
ICLOCK
PRWT
SERCHI
STEPEN
WATEST

Commons used:

CCDE
CONST
DESCRP
ERROR
GRAPH
INPUT
WINDP

DESPC

This subroutine modifies the final portion of the cruise and descent profile to include the sink rate constraint HDOTC from cruise altitude down to altitude HCABSL. DESPC first determines where the descent crosses HCABSL. It then generates a new, constrained upper portion of the descent with a ramped Mach number and constant sink rate; it modifies the descent table accordingly. Finally, DESPC calls VOPTRJ to join climb and descent with a new cruise segment.

Subroutines called:

ATLOW
DESCND
FIASM
FOPT
ICLOCK
TRIM
VOP
VOPTRJ
WIND

Commons used:

CCDE
CONST
CRUISE
DESCRP
GRAPH
ERROR
INPUT
WINDP

DRAGC

This function computes the drag coefficient C_D as a function of Mach number and lift coefficient C_L , using subroutine CDRAG and the weight, aircraft and atmospheric parameters previously stored in common. It assumes lift equals weight to compute C_L .

Subroutines called:

CDRAG

COMMONS Used:

CCDE
DESCRP
INPUT

ENGEP

This subroutine calls the appropriate routine to compute the aircraft maximum thrust and EPR, the thrust associated with the input EPR, and the fuel flow rate. The engine model is associated with the particular aircraft model selected by the input variable IAC. Currently, two models are available, but logic is presented to use up to four different aircraft.

Subroutines called:

ENGEP1*
ENGEP2
ENGEP3
ENGEP4*

Commons used:

None

* not included in program

ENGEP2

This subroutine generates the thrust THRUST and the fuel flow rate FDOT for some given altitude H, Mach number EMAKNO and EPR setting. First EPRMX, the maximum continuous EPR, is determined by table look-up for some given temperature Ta and altitude H, where

$$T_a = T(1 + \frac{\gamma-1}{2} (EMAKNO)^2)^2 - 273.15.$$

Here, T is the temperature corresponding to altitude H, after temperature variation correction, and

$$\gamma = 1.4, \text{ the ratio of specific heats.}$$

The EPR setting is limited to $EPR \leq EPRMX$ for cruise and $EPR \leq EPRMX - .1$ for climb or descent.

Second, (FN/δ_e) is computed from a polynomial. Then, the thrust is computed as,

$$THRST = 3(\delta_{am}) (FN/\delta_e).$$

This is the thrust for the medium range tri-jet transport aircraft model. Here, δ_{am} is the pressure ratio

$$\delta_{am} = \frac{P}{P_o}.$$

Here, P is the atmospheric pressure corresponding to altitude H, and P_o is the atmospheric pressure at sea level. A factor of three is used since there are three engines.

Finally, the fuel flow rate FDOT is computed as:

$$FDOT = 3 * WFC * \delta_a * K_c$$

where

$$K_c = .00223181 T_a + .9675897,$$

$$\delta_a = \delta_{am} (1 + \frac{\gamma-1}{2} (EMAKNO)^2)^{\gamma/\gamma-1}.$$

Also, WFC is the fuel-flow rate computed as a polynomial of EPR, where the coefficients of the polynomial depend on both altitude and Mach number.

Subroutines called:

ATLOW
DBLSRC
POLYE1

Commons used:

CCDE
ERROR
INPUT
TRIJET

ENGEP3

This subroutine generates the thrust THRUST and the fuel flow rate FDOT for some given altitude H, Mach number EM and EPR setting. First EPRMX, the maximum continuous EPR, is determined by table look-up, (Tables 6 and 7 in Block Data) for some given temperature Ta and altitude H, where

$$T_a = T(1 + \frac{\gamma-1}{2} (EM)^2) - 273.15.$$

Here, T is the temperature corresponding to altitude H, after temperature variation correction, and

$$\gamma = 1.4, \text{ the ratio of specific heats.}$$

The EPR setting is limited to $EPR \leq EPRMC$.

Second, (FN/δ_e) is computed from a polynomial. Then, the thrust is computed as,

$$THRST = 2(\delta_{am}) (FN/\delta_e),$$

where δ_{am} is the pressure ratio

$$\delta_{am} = \frac{P}{P_o}.$$

Here, P is the atmospheric pressure corresponding to altitude H, and P_o is the atmospheric pressure at sea level. A factor of two is used since there are two engines.

Finally, the fuel flow rate FDOT is computed. A polynomial is used to calculate WFC for a given EPR and altitude. At values of $EPR < 1.6$, there is also a correction for Mach number (Table 10 in Block Data). Then,

$$FDOT = 2 * WFC * \delta_a * K_c,$$

where

$$K_c = .0022 T_a + 0.97,$$

$$\delta_a = \delta_{am} (1 + \frac{\gamma-1}{2} (EM)^2)^{\gamma/\gamma-1}.$$

Subroutines called:

DBLSRC
POLY2
SGLSRC

Commons used:

CCDE
ERROP
INPUT
TWINJT

ENGIDL

ENGIDL is called during descent to compute thrust and fuel flow rate for idle EPR. It does this through table look-up for the appropriate aircraft.

Subroutines called:

DBLSRC
SGLSRC

Commons used:

ERROR
TRIJET
TWINJT

ESTCD

This subroutine calls the appropriate routine to estimate cruise range before descent based on the particular aircraft model selected by the input variable IAC. Currently, two models are available, but logic is present to use up to four different aircraft.

Subroutines called:

ESTCD1*
ESTCD2
ESTCD3
ESTCD4*

Commons used:

None

* not included with program

ESTCD2

This subroutine estimates cruise range for the tri-jet, using a polynomial derived from experience with the trajectory computation. For the optimal profile, this polynomial gives a value of 115 nautical miles. For suboptimal profiles, it is less.

Subroutines called:

POLYE1

Commons used:

DESCRP

ORIGINAL PAGE IS
OF POOR QUALITY

ESTCD3

This subroutine estimates cruise range for the twin-jet. For an optimal flight, it estimates this range to be the difference between total range-to-go and climb plus the estimated descent range of 90 miles. For the suboptimal case, the cruise range is computed as a function of cruise energy and $d\lambda/d(\text{energy})$.

Subroutines called:

SGLSRC

Commons used:

CCDE
CRUISE
GRAPH
DESCRP
INPUT

ESTDF

This subroutine calls the appropriate routine to estimate the descent fuel based on the particular aircraft model selected by the input variable IAC. Currently, two models are available, but logic is present to use up to four different aircraft.

Subroutines called:

ESTDF1*
ESTDF2
ESTDF3
ESTDF4*

Commons used:—

None

* not included with program

ESTDF2

This subroutine estimate descent fuel for the tri-jet, using an empirically determined polynomial.

Subroutines called:

None

Commons used:

DESCRP

ESTDF3

ESTDF3 estimates descent fuel for the twin-jet model as a function of the difference in energy between top of descent and landing. Different empirically derived functions are used depending on whether the single control or dual control case is being computed.

Subroutines called:

POLY2

Commons used:

CCDE
CONST
DESCRP
INPUT

ESTEP

This subroutine computes the next energy step size during climb and descent. The nominal step is DENRGY, but this may be reduced when within 3000 feet of the estimated cruise energy. ESTEP also sets the quantities NEARCZ (within 5000 feet of cruise) and IOPT (two-control flag only).

Subroutines called:

None

Commons used:

CCDE
DESCRP
INPUT

FBOUND

This function evaluates the drag force D in lb (using function DRAGC) if IDRAG is set equal to 1, and the absolute value of maximum thrust (TMAX) minus drag if IDRAG is equal to 2. To determine the maximum thrust, it calls the subroutine ENGEPR with thrust setting TMAX set equal to its maximum value.

The output is:

$$\begin{aligned} \text{FBOUND} &= D \text{ in lb, if IDRAG} = 1 \\ &= |TMAX - D|, \text{ if IDRAG} = 2 \end{aligned}$$

Subroutines called:

DRAGC
ENGEPR

Commons used:

CCDE
DESCRP
ERROR
INPUT

FCLIMB

This function evaluates the Hamiltonian for climb and descent for given values of airspeed VTAS, energy E, and EPR setting. The altitude is obtained by

$$H = E - V^2/2g.$$

The atmospheric parameters and the aircraft ground speed V_g are found. The drag D is obtained by calling function DRAGC with Mach number and aircraft weight as inputs. The Hamiltonian is then evaluated by calling FTHRST.

Subroutines called:

ATLOW
DRAGC
FTHRST

Commons used:

CCDE
CONST
DESCRP
ERROR
INPUT

FCLMB6

This function minimizes the Hamiltonian for climb. First SPLMT is called to calculate speed limits at the given altitude, and then the appropriate permissible region for minimization is computed. Then FCLMB6 calls FMIN and FCLIMB for minimization over speed. It also calls FMIN and FTHRST for minimization over EPR setting, if required for the two-control case (IUPI=1).

Subroutines called:

FCLIMB
FDRAG
FMIN
FTHRST
PILIMT
SPLMT

Commons used:

CCDE
DESCRP
ERROR
INPUT

FCOST

This function evaluates the cost of flight per nautical mile, in dollars (cost of fuel and time) when IMFD is set equal to 0. It determines fuel flow rate in lb/hr if IMFD is set equal to 1. To do so, FCOST calls first TRIM to obtain the trim condition for constant speed level flight at a given altitude H and the Mach number EM.

Subroutines called:

TRIM
WIND

Commons used:

CCDE
DESCRP
ERROR

INPUT
WINDP

FDRAG

This subroutine computes the drag D for some given true airspeed VTAS, energy E and aircraft weight W if IDRAG = 1. It computes T - D, if IDRAG \neq 1. Function DRAGC is used in computing D. The subroutine ENGEPR is called for computing T.

Subroutines called:

ATLOW
DRAGC
ENGEPR

Commons used:

CCDE
DESCRP
ERROR
INPUT

FDSCN6

This function minimizes the Hamiltonian for descent. First SPLMT is called to calculate the speed limits at the given altitude, and then FDSCN6 calls FMIN and FCLIMB for minimization over speed. In the single-control case, this is all that is required. In the two-control case, FDSCN6 goes on to minimize over EPR setting as well.

Subroutines called:

~~FCLIMB~~
FDRAG
FMIN
FTHRST
PILINT
SPLMT

Commons used:

CCDE
DESCRP
ERROR
INPUT

FIAS

FIAS returns Mach number as a function of indicated airspeed (in feet per second) and atmospheric pressure.

Subroutines called:

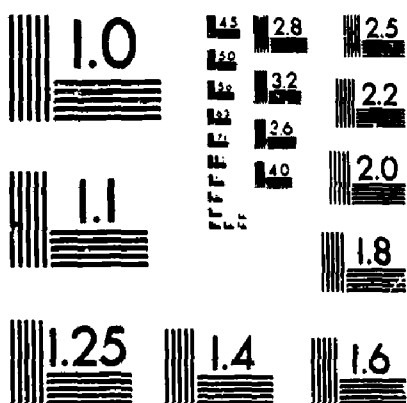
None

Commons used:

None

2 OF 2

30061 UNC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

FIASM

FIASM returns indicated airspeed in knots as a function of Mach number and atmospheric pressure.

Subroutines called:

None

Commons used:

CONST

FMIN

This function minimizes the unimodal function $F(X)$ so that x is within $1/144$ of the range between input boundaries by a Fibonacci search, where 144 is the eleventh Fibonacci number.

FMIN2

This function minimizes the unimodal function $F(X)$ so that x is within $1/34$ of the input range by a Fibonacci search where 34 is the eighth Fibonacci number.

NOPT

This function computes the optimal airspeed for minimum cruise cost if IMFD is set equal to 0. It computes optimal airspeed for minimum fuel rate if IMFD is set equal to 1. First the minimum drag DRGMN is computed by calling FMIN. Then, the maximum thrust is computed by calling ENGKPR. Third, the lower and upper bounds FA and FB, which define the range of permissible Mach numbers are computed by calling FMIN twice, using FBOUND for evaluation. Finally the optimum Mach number OPTMAK and the minimum cost are computed by the function FMIN, using FCOST for evaluation.

Subroutines called:

ATLOR
DRAGC
ENGKPR
FBOUND
FCOST
FMIN

Commons used:

CODE
DESCR
ERROR
INPUT
WINDP

FMKST

This function evaluates the Hamiltonian for some given FPR setting, Mach number EMACH, true airspeed VTAS, altitude HALT, drag D and weight W. The values of thrust F and fuel flow rate FF are obtained from calling the subroutine ENGKPR.

Subroutines called:

ENGKPR
ENGIM
WIND

Commons used:

CODE
CONST
DESCR
ERROR
INPUT
WINDP

FULEST

FULEST calls the appropriate routine to estimate the climb fuel use for the particular aircraft model selected by the input variable IAC. Currently, two models are available, but logic is present to use up to four different aircraft.

Subroutines called:

FULEST1*
FULEST2
FULEST3
FULEST4*

Commons used:

None

* not included with program

FULEST2

This subroutine is used to estimate the climb fuel and the aircraft weight WCRUZ at the beginning of cruise for the tri-jet aircraft. WCRUZ is needed to interpolate the cruise table to obtain initial cruise values of energy, altitude, and cost.

The routine uses cost of fuel FC, cost of time TC, initial weight WTO, and cruise energy minus initial energy to estimate fuel.

Subroutines called:

WLEFHV

Commons used:

CCDE
DESCRP
ERROR
INPUT

FULST3

This subroutine is used to estimate the climb fuel and the aircraft weight WCRUZ at the beginning of cruise for the twin-jet aircraft. The routine uses cost of fuel FC, cost of time TC, initial weight WTO, and cruise energy minus initial energy to estimate fuel.

Subroutines called:

POLY2
WLEFHV

Commons used:

CCDE
DESCRP
ERROR
INPUT

ICLOCK

This subroutine resolves TIME in seconds into hours IHR, minutes IMIN and seconds ISEC. For example, 4521 seconds is resolved into 1 hour 15 minutes and 21 seconds.

JTRUNC

This subroutine finds the last point of a monotonically decreasing series.

LSQPOL

This subroutine makes a least square polynomial fit for y as a function of x. The arrays x(.), y(.) contain corresponding points of the independent variable and the value of the function. The total number of points in the set is specified by N, the degree of polynomial to be fitted is specified by M - 1 (e.g. M = 3 for quadratic fit), and the coefficients of the polynomial are stored in the array B(.) in the following order:

$$y = B(3)x^2 + B(2)x + B(1).$$

LSQPOL uses the subroutine MATINV. Common used is ANE206

MATINV

This subroutine inverts a matrix A and stores the result in A. The dimension of the matrix is specified by M. The common used is ANE206.

NICER

This subroutine takes minimum and maximum values of the optimum trajectory variables and computes appropriate boundaries for the printer plots.

NICER uses no other subroutines.

PAGE

This subroutine advances the printout to the top of the next page.

PCCMP5

This subroutine computes P , the percentage of variation from ψ_{opt} . The value of P is computed by a local polynomial fit to the ψ versus R curves which have two points to begin (R_{min} , 1.5ψ or 1.3ψ) and (R_{max} , 1.01ψ or 1.0ψ). Subsequent iteration for synthesizing the fixed range trajectory increases the number of points. The value of R and its corresponding percentage of variation are stored in the arrays $RANCF$ and C respectively. The total number of points at any iteration is specified by IPC .

Subroutines called:

FULEST
MATINV
WCLST

Commons used:

CCDE
DESCRP
ERROR
GRAPH
INPUT

PICTUR (k, x, y, IS)

This subroutine generates the printer plots and is called $n + 2$ times (for n points to be plotted) with k less than, equal to, or greater than zero, depending on the purpose of the call.

First Call: $k < 0$. PICTUR stores one (x,y) point in the plot array and sets up the plot axes and scales.

Next n Calls: $k = 0$. PICTUR stores one (x,y) point in the plot array using the character designated by the parameter IS .

Last Call: $k > 0$. PICTUR prints out the plot and writes the labels.

Subroutines called:

None

Commons used:

GRAPH

PILIMTORIGINAL PAGE IS
OF POOR QUALITY

This function evaluates the absolute value of $T - D$ for some given EPR setting, altitude, Mach number and drag. It imposes a heavy penalty if the function is outside the boundary. Specifically, this penalty occurs for climb if it is outside the region where $T < D$. The penalty occurs for descent, if it is in the region where $T > D$.

Subroutines called:

ENGEPR

Commons used:

CCDE
DESCRP
ERROR
INPUTPOLY1

This function evaluates the polynomial

$$Y = b(1) + b(2) X + b(3) x^2 = \dots b(M) X^{m-1}$$

POLY2

POLY2 evaluates the polynomial

$$\begin{aligned} Z = & c_{11} + c_{12}x_2 + \dots c_{1m}x_2^{n-1} \\ & + c_{21}x_1 + c_{22}x_1x_2 + \dots c_{2n}x_1x_2^{n-1} \\ & + \dots \\ & + c_{m1}x_1^{m-1} + c_{m2}x_1^{m-1}x_2 + \dots + c_{mn}x_1^{m-1}x_2^{n-1} \end{aligned}$$

PRETBL

PRETBL prints out the cruise performance table for the lower cruise segment of a step climb flight. It uses the lower cruise table which was saved previously by subroutine STEP.

Subroutines called:

FIASM
ICLOCK
SERCH1
WIND

Commons used:

CCDE
CRUISE
CONST
DESCRP
ERROR
GRAPH
INPUT
TOA
WINDP

PRFTBL

This subroutine writes the optimum climb and descent trajectory variables. It also writes the cruise performance table. In a step climb run, PRFTBL calls PRETBL to write the lower cruise segment and STEPUP to write the step climb trajectory.

When the input quantity IGRAF is non-zero, this routine writes a dataset on Unit 11 for storage by the user. This dataset may be used for subsequent graphing. If IGRAF is greater than one, PRTPLT is called to produce printer plots.

Subroutines called:

CRUISR
CRUISX
FIASM
ICLOCK
PAGE
PRETBL
PRTPLT
STEPUP
WIND

Commons used:

CCDE
CRUISE
CONST
DESCRP
ERROR
GRAPH
INPUT
WINDP

PROFIL

This subroutine controls the trajectory calculations. For a normal entry, it calls CRZOP5 to calculate the cruise table and then calls PRSUM to print out the cruise summary table. PROFIL then goes through the following sequence.

1. If this is a two-part trajectory, go to step 2. Otherwise, call CLIMB to generate the climb trajectory.
2. If this is a constrained descent, go to step 9.
3. Call DESCND to synthesize the descent profile, based on an estimated landing weight.
4. Call VOPTRJ to estimate the cruise distance and fuel and to revise the landing weight.
5. Call DESCND to generate the refined descent trajectory.
6. Call VOPTRJ to generate the overall trajectory, including the total ground track distance covered by the trajectory.
7. If this is a three-part, free altitude, non time-of-arrival run, call PCCMP5 to test this distance against the desired range R. If PCCMP5 returns with flag IRETRN, indicating that the trajectory has either been synthesized or cannot be, PROFIL returns to its calling program. If the trajectories to determine R_{\min} and R_{\max} have not been synthesized, return to Step 1. If the required distance is within the interval (R_{\min}, R_{\max}) , return to Step 1. Otherwise if R is greater than RMAX, return to step 4.
8. If PCCMP5 is not called, PROFIL simply returns.
9. If a constrained descent is desired, call DESPC to calculate a new descent, and then return.

Subroutines called:

CLIMB	PCCMP5
CRZOP5	PRSUM
DESCND	VOPTRJ
DESPC	

Commons used:

CCDE	ERROR
CONST	INPUT
CRUISE	
DESCRP	

PRSUM

PRSUM calculates and prints out the cruise summary table. It also calculates the summed cruise time and distance tables.

Subroutines called:

ATLOW
PAGE
WIND

Commons used:

CCDE
CRUISE
DESCRP
ERROR
INPUT
WINDP

PRTPLT

The subroutine calls subroutine PICTUR to generate printer plots of the optimum trajectory variables when the flag IGRAF is set greater than 1 in the subroutine PRFTBL. Labels to appear on plots are stored in array NOTES (8) which provides eight lines of ten characters each. The maximum and minimum values are calculated in subroutine PRFTBL and scaled in subroutine NICER.

For IGRAF = 2, Mach no, flight path angle, altitude and fuel are plotted versus range.

IGRAF > 2, all variables are plotted versus range and time.

Subroutines called:

NICER
PICTUR

Commons used:

GRAPH
INPUT

PRWT

This subroutine prints the estimated conditions at top of climb.

Subroutines called:

None

Commons used:

CCDE
CONST
ERROR

SERCHD

The array TX(.) is monotonically decreasing. This subroutine searches the index x such that

$$TX_{\ell} \geq x \geq TX_{\ell+1} ,$$

and returns both ℓ and pf where

$$pf = \frac{TX_{\ell} - x}{TX_{\ell} - TX_{\ell+1}} .$$

SERCHI

The array TX(.) is monotonically increasing. This subroutine searches the index ℓ such that

$$TX_{\ell} \leq x \leq TX_{\ell+1} ,$$

and returns both ℓ and pf where

$$pf = \frac{x - TX_{\ell}}{TX_{\ell+1} - TX_{\ell}} .$$

SGLSRC

This function evaluates a single function F at the point A . This is done by linear interpolation to obtain A 's location in the array X and using the tabulated values of $F(X)$.

Subroutines called:

SERCHI

Commons used:

None

SPLMT

SPLMT controls the speed limits for the aircraft during climb and descent. Different limits are applied during flight in the lowest 10000 feet and the last 3000 feet. Limits also depend on the type of aircraft.

Subroutines called:

ATLOW
FIAS
SGLSRC

Commons used:

CCDE
CONST
DESCRP
ERROR
INPUT

STEP

STEP controls the logic for the step climb option. It is called (by either OPTM56 or OPTTOA) after an initial optimum profile is computed. STEP then saves the initial cruise table and calls CRZOP5 to calculate a new cruise table at an altitude 4000 feet higher.

Function FMIN2 is then used to find an optimum trajectory consisting of a lower cruise segment, a step climb, and an optimum two-part cruise and descent segment. (See the description of STEPOPT for details of the function being minimized.) The new trajectory is only accepted if its total cost is less than the flight profile without the step climb.

Subroutines called:

CRUISR
CRZOP5
FMIN2
PRSUM
SERCHD
STEPOPT
STEPUP

Commons used:

CCDE
CRUISE
DESCRP
ERROR
INPUT
GRAPH
TOA

STEPEN

Subroutine STEPEN computes the trajectory variables associated with a single energy step. It finds the altitude at the new energy level, updates the weather with ATLOW, finds the fuel flow from ENGEPR, and calculates time, distance, velocity, and flight path angle.

Subroutines called:

ATLOW
ENGEPR
FIAS
FIASM
WIND

Commons used:

CCDE
CONST
DESCRP
ERROR
INPUT
WINDP

STEPOPT

This function calculates a step climb trajectory. Input (from FMIN2) is the distance to be flown in the lower cruise table. STEPOPT then cruises this distance, calls STEPUP for a ramped Mach number climb at full power, and then calls PROFIL to compute an optimum two-part trajectory beginning at the top of climb. The final value of the function is the total cost of the four part trajectory consisting of lower cruise, step climb, upper cruise and descent.

Subroutines called:

CRUISR
PROFIL
SERCHD
SERCHI
STEPUP
WLEFHV

Commons used:

CCDE
CRUISE
DESCRP
ERRCR
INPUT
TOA

STEPUP

This subroutine computes a ramped Mach number climb at full power. The climb is from altitudes HALT to HCRU in steps of DELTAH. —

Subroutines called:

ATLOW
DRAGC
ENGEPR
FIASM
WIND
—

Commons used:

CCDE
CONST
CRUISE
DESCRP
ERROR
INPUT
TOA
WINDP

TRACIT

In case of error, this subroutine provides a "walk back" through the subroutine calling hierarchy. If the subroutine is set up to recognize the computation or logic to be in error, then TRACIT can be used to find the source of the error.

ORIGINAL PAGE IS
OF POOR QUALITY

TRIM

This subroutine is used to compute the trim conditions for medium range transport aircraft. This subroutine computes angle-of-attack α and thrust T to keep the aircraft in trim for constant speed level flight, for a given altitude and for a given Mach number.

With γ the flight path angle, the equations of motion in the horizontal and vertical directions are as follows:

$$\frac{W}{g} (dv/dt) = T \cos \alpha - D - W \sin \gamma$$

$$\frac{W}{g} v(d\gamma/dt) = T \sin \alpha + L - W \cos \gamma$$

For a trimmed condition:

$$(dv/dt) = (d\gamma/dt) = 0.$$

The two equations are combined to eliminate thrust to give the equation:

$$(W \cos \gamma - 1) \cos \alpha - (D \sin \alpha + W \sin \gamma) \sin \alpha = 0.$$

This equation is solved by iterating with angle-of-attack, α .

Once the aircraft is trimmed, the thrust is solved from the drag

$$T = (D + W \sin \gamma) / \cos \alpha.$$

This required thrust is matched by iterating on values of power setting (EPR) and calling subroutine ENGEPR. Once the correct power setting is determined, the engine fuel flow is also known.

Subroutines called:

CDRAG
CLIFTT
ENGEPR

Commons used:

CCDE
DESCRP
ERROR
INPUT

VOPTRJ

This subroutine (1) computes the climb fuel FCLMB, climb time TCCLMB, climb distance DCLMB, descent fuel FDOWN, descent time TDOWN, and descent distance DDOWN; (2) cruise fuel FCRULB, cruise distance DCRUZ, and cruise time TCRUZ; (3) the final cruise weight WCRUZ; (4) the cruise fuel efficiency EFCRUZ (lb/n.mi.) and overall fuel efficiency EFFCNZ (lb/n.mi.), and (5) the landing weight WLNDR. It also generates the fuel used in lb, the distance covered in miles, the time duration in hr: min: sec, the total cost in \$, and the cost per nautical mile for climb, cruise, descent, and the overall trajectory.

If VOPTRJ is called during a step climb optimization, it includes the lower cruise segment and step climb time, fuel and distance in these totals. VOPTRJ also prints out the initial cruise weight, final cruise weight, and their corresponding true airspeed, cruise cost, equivalent airspeed, cruise energy, ground speed, altitude and Mach number. Finally, it prints out the cruise fuel efficiency, the overall fuel efficiency, and the landing weight.

Subroutines called:

ATLOW	SERCHI
CRUISX	WIND
FLASM	WIEFHV
PAGE	WRITE1

Commons used:

CCDE	ERROR
CONST	GRAPH
CRUISE	INPUT
DESCRP	WINDP

WATEST

This subroutine is used to estimate the cruise range, fuel burned during descent (FDOWN) and the aircraft landing weight WLNDC. For free cruise altitude optimization using V and n , ($IVP1 = 1$) the cruise range is zero. For optimization using V only, the cruise distance is estimated using ESTCD. For fixed cruise altitude, the cruise distance is assumed to be the range minus the climb distance minus 90. CRUISX is called to bring the flight parameters to this range (estimated top of descent). Then WLEFHV is called to estimate cruise fuel rate and ground speed.

For all cases, ESTDF is then called to compute the descent fuel. The landing weight is then calculated from climb, cruise, and descent fuel use.

Subroutines called:

CRUISX
ESTCD
ESTDF
WLEFHV

Commons used:

CCDF
CONST
DESCRP
ERROR
GRAPH
INPUT
TOA

WCLST

This subroutine is used to compute climb, cruise, and descent segments for the fixed thrust case ($IVP1 = 0$) where the value of cruise cost is very close to the optimum value. This routine is called from PCOMP5.

Subroutines called:

CLIMB
DESCND
DESPC
VOPTERJ (at entry point VOP)

Commons used:

CCDF
CONST
DESCRP
ERROR
GRAPH
INPUT

WIND

This subroutine computes the wind velocity as a function of altitude. This is combined with the aircraft velocity with respect to the air mass to compute ground velocity. Inputs to this program are H, the altitude in feet; PSIG, the aircraft heading in degrees; VTAS the aircraft air speed; GAMMR, the angle of attack; VW, and PSIW arrays. The outputs from this program are VWA, the wind speed, and VG, the aircraft ground speed.

Subroutines called:

SERCHI

Commons used:

DESCRP
ERROR
INPUT
WINDP

WINDIN

This subroutine reads in the wind profile (the magnitude and direction of wind as a function of altitude). The wind magnitude input is in knots and the program converts it to ft/sec and stores it in the VW array. The wind direction is stored in PSIW in degrees. The input represents the direction the wind is coming from. The program adds 180° to this value to obtain the vector direction. The altitudes corresponding to these wind magnitudes and directions are stored in array HWIND. The wind may be input as a single profile valid over the entire flight, or as separate climb, cruise and descent profiles. In the case of a step climb the cruise profile is used for lower cruise, step climb, and upper cruise segments.

Subroutines called:

None

Commons used:

INPUT
WINDP

WLEFHV

The purpose of this subroutine is to interpolate cruise table results to obtain cruise cost ELAMBS, cruise energy ECRUZ, cruise fuel flow rate FCRUZ, cruise altitude HCRUZ, and cruise airspeed and ground speed VCKTAS and VGKNT as functions of the input cruise weight WCRUZ.

Subroutines called:

CTABLE
JTRUNC
SERCHD
WIND

Commons used:

CCDE
CONST
CRUISE
DESCRP
ERROR
INPUT
WINDP

WRITE1

This subroutine (a) computes the climb, cruise, descent, or overall cost for flying the specified segment of the trajectory, where the cost is the sum of fuel cost and time cost, (b) computes the cost per nautical mile to fly, (c) resolves the time given in seconds into hours, minute and seconds, and (d) writes out the description of the segment (i.e. climb, cruise, descent, total), the fuel used, the distance traversed, the time duration (in hours, minutes, seconds), the cost for the segment, and the cost per nautical mile to fly.

Subroutines called:

ICLOCK

Commons used:

CONST
INPUT

REFERENCES

1. Anon., "TRAGEN - Computer Program to Simulate an Aircraft Steered to Follow a Specified Vertical Profile; User's Guide," NASA CR-166062, March 1983.
 2. Erzberger, H., and Lee, H.Q., "Characteristics of Constrained Optimum Trajectories with Specified Range," NASA TM-78519, 1978.
 3. Lee, H.Q., and Erzberger, H., "Algorithm for Fixed Range Optimal Trajectories," NASA TP-1565, July 1980.
 4. Sorensen, J.A., "Concepts for Generating Optimum Vertical Flight Profiles," NASA CR-159181, September 1979.
 5. Sorensen, J.A., and Waters, M.H., "Generation of Optimum Vertical Profiles for an Advanced Flight Management System," NASA CR-165674, March 1981.
 6. Sorensen, J.A., Waters, M.H., and Patmore, L.C., "Computer Programs for Generation and Evaluation of Near-Optimum Vertical Flight Profiles," NASA CR-3688, May 1983.
 7. Bryson, A.E., and Ho, Y.C., Applied Optimal Control, Blaisdell, Waltham, Mass., 1969.
 8. Bryson, A.E., Desai, M.N., and Hoffman, W.C., "Energy-State Approximation of Supersonic Aircraft," J. of Aircraft, Vol. 6, No. 6, Nov.-Dec. 1979.
-